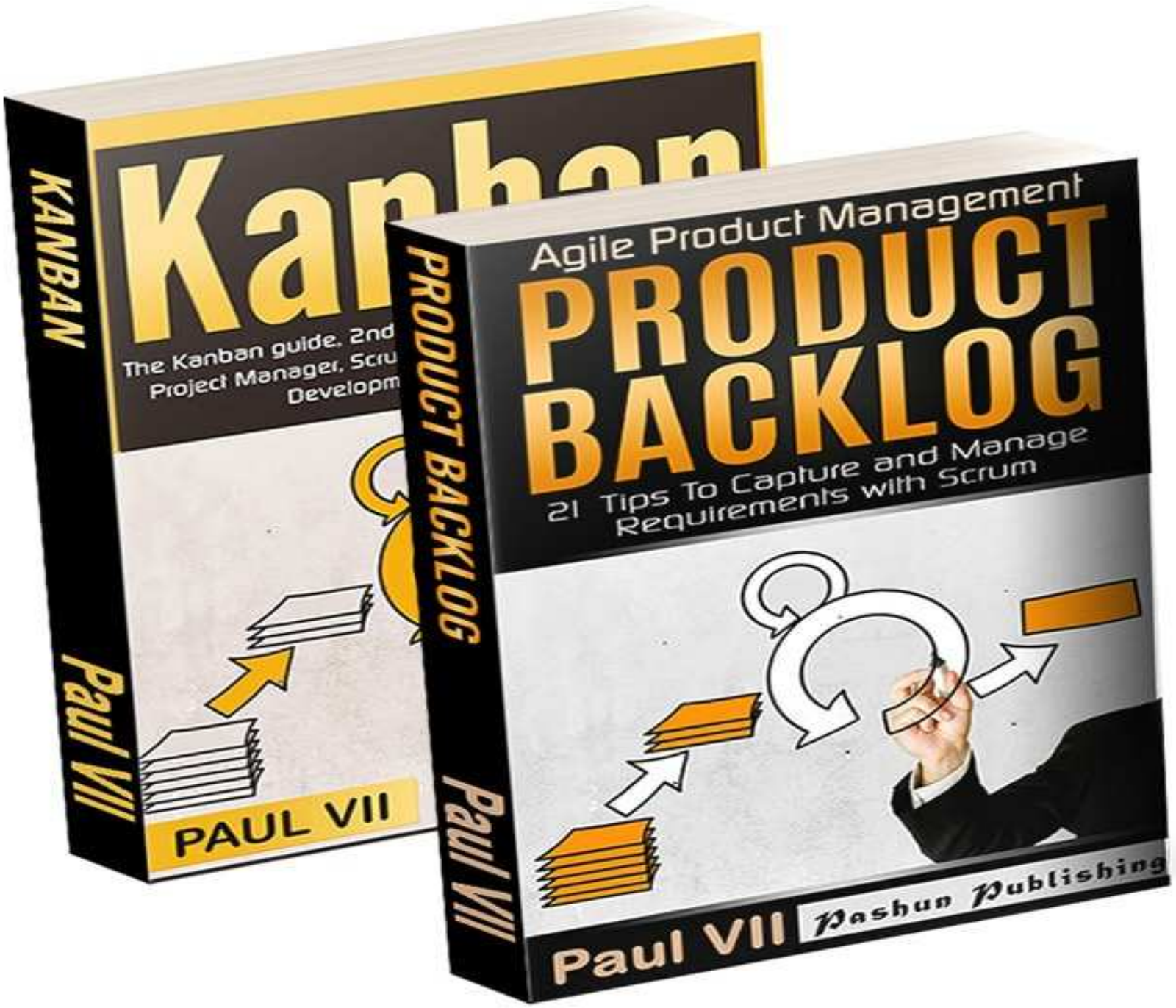

KANBAN
Kanban
The Kanban guide, 2nd
Project Manager, Scru
Developm
PAUL VII
Paul VII
PRODUCT BACKLOG
Agile Product Management
PRODUCT BACKLOG
21 Tips To Capture and Manage Requirements with Scrum
Paul VII
Pashun Publishing
Paul VII

Kanban:

The Kanban guide, 2nd Edition:

For the Business, Agile Project Manager, Scrum Master, Product Owner and Development Support Team

Introduction

Thank you and congratulations on taking this class, "*Kanban: The KanbanGuide 2nd Edition: For the Business, Agile Project Manager, Scrum Master, Product Owner and Development Support Team*".

Creating the most efficient system for your business, development team, support team or any other organisation can often be a game of trial an error. We spend years aiming to create efficient practices which meet the demands of our customers as quickly as possible without overspending our effort, time or money. Imagine if there was a system that helped optimise the effort of our teams to meet the demands of our customers.

In this book, you will learn how the Kanban system can do this, and I explain:

- What Kanban is and where it originates from
- How it can make your team more efficient
- How to implement Kanban in your team or organisation
- Exercises to plan a Kanban System for your team or organization
- An overview of Scrum-ban - a technique used for combing the Scrum framework with Kanban especially for project based environments
- A brief overview of Agile Scrum

When you have read this guide, you will have an excellent foundation in thispopular system, and a great basis for helping others to understand the same.

© Copyright 2016 by Pashun Publishing Press - a division of Pashun Consulting Ltd. - All rights reserved.

This document is geared towards providing exact and reliable information in regards to the topic and issue covered. The publication is sold with the idea that the publisher is not required to render accounting, officially permitted, or otherwise, qualified services. If advice is necessary, legal or professional, a practiced individual in the profession should be ordered.

- From a Declaration of Principles which was accepted and approved equally by a Committee of the American Bar Association and a Committee of Publishers and Associations.

In no way is it legal to reproduce, duplicate, or transmit any part of this document in either electronic means or in printed format. Recording of this publication is strictly prohibited and any storage of this document is not allowed unless with written permission from the publisher. All rights reserved.

The information provided herein is stated to be truthful and consistent, in that any liability, in terms of inattention or otherwise, by any usage or abuse of any policies, processes, or directions contained within is the solitary and utter responsibility of the recipient reader. Under no circumstances will any legal responsibility or blame be held against the publisher for any reparation, damages, or monetary loss due to the information herein, either directly or indirectly.

Respective authors own all copyrights not held by the publisher.

The information herein is offered for informational purposes solely, and is universal as so. The presentation of the information is without contract or any type of guarantee assurance.

The trademarks that are used are without any consent, and the publication of the trademark is without permission or backing by the trademark owner. All trademarks and brands within this book are for clarifying purposes only and are the owned by the owners themselves, not affiliated with this document.

Table of Contents

Kanban:

Chapter 1. Introducing Kanban

Chapter 2. The Kanban Board - Deciding on your Workflow

Chapter 3. Determining Priority

Chapter 4. Understand The “ Pull System ”

Chapter 5. Establish Work In Progress Limits

Chapter 6. Measure and Manage Flow

Chapter 7. Making Policies and Processes Explicit

Chapter 8. Create and Refining your Process

Chapter 9. Using Models to recognize improvement opportunities

Chapter 10. Using Kanban with Scrum

Preview Of ‘ The Scrum Master Mega Pack ’

Check Out My Other Books

Bonus: Subscribe to Download the FreeScrumEbook and Bonuses

Chapter 1. Introducing Kanban

What is Kanban?

Kanban is a simple scheduling system used to achieve Just In Time (JIT) delivery of products. Although the Kanban system has more recently become a favourite in the software business, you will learn that its origin's lie far beyond the realms of software. Using this system balances the demand of our customers against the very real limits of our workforce. It also allows companies and teams to achieve a finely tuned balance between supply and demand, issues and maintenance, sales and orders.

The Kanban System

We first define a *workflow,* which is specific to our business. We then use a system of *priority* setting, to decide which pieces of work should be tackled first. Following this, we use the concept of a *pull system* which means that each item of work is pulled from a queue by whoever is doing the work as opposed to being pushed onto that person's stack. *Work In Progress limits* are set to avoid overloading any of our workers and *core rules* are established and agreed to keep the system in check. Throughout all of this, we constantly *refine* the system and fine-tune it to improve efficiency. I will explain each of these concepts in more detail, but first, let us understand where this system came from and how it has been proven to work.

Japanese Kanban

The Japanese word “kanban”, loosely translates to "signboard" in English. This relates to the signboards that were used to direct: people to the shelves to buy products; workers to replenish stock; and eventually companies to order more stock.
Toyota first developed this concept in the late 1940s. Although then they were an automobile company, they were keen to learn from other industries such as supermarkets. These studies are what led to the well- known Just In Time (JIT) concept. Creator of Kanban, Taiichi Ohno learned that supermarkets achieved efficiency as follows.

- Customers buy products from the supermarket floor
- This signals shop workers to get stock from the stock room
- In turn this signals the stock room to order more stock from suppliers

✓ the supplier then replenishes the stock room,

The benefit of this method was (and still is):

- the supermarket stocks just enough to replenish the shelves
- the shelves are only restocked when there is a gap to be filled

1. the customers always have some of the product available when they need it

As previously mentioned, sign boards were used as triggers to signal where the product could be bought and when restocking was necessary. Therefore the name signboard or “kanban” was born, along with a simple system.

Origin of Kanban for development

In the development world, Kanban first shot to fame in the formof the method described by David J Anderson. There is a hugeamount on Kanban in his book *Kanban: Successful EvolutionaryChange for Your Technology Business*. The five core properties he describes are:

- Visualise the workflow: This makes it easy to see howwork is progressing..

- Limit work-in-progress: Therefore there is a limit to theamount of work in process at any one time, (aided by apull system)

- Measure and manage flow: This means constant reevaluationof the workflow so we can change it whennecessary.

- Make process and policies explicit: i.e. there should beagreement and consensus on working policies

- Use models to recognise improvement opportunities:There should be methods to review and communicateissues with or views on the workflow. Those involvedwork together to find the best way to move forward usingscientific methods.

Key Points

The main thing to focus on in this section is the 5 core properties of Kanban for development as set out by David. J. Anderson. These are the guiding properties that can be used to carry out Kanban in any business. Therefore, the main body of this class is split into a different section for each property. So here is a re-cap. Understand these and you have an excellent foundation in Kanban.

Each property will be explained in more detail so don't worry if you don't understand some of the terms fully!

5 Core Properties

o *Visualise the workflow:* This makes it easy to see how
work is progressing and where there are issues to solve to make progress.

o *Limit work-in-progress:* Therefore there should be a limit to the
amount of work in process at any one time to prevent overloading of the team(s).

o *Measure and manage flow:* This means constant reevaluation
of the workflow so we can change it whennecessary to improve the amount of work delivered.

o *Make process and policies explicit:* i.e. there should be
agreement and consensus on working policies within the organisation.

o *Use models to recognise improvement opportunities:*
There should be methods to review and communicate
issues with or views on the workflow. Those involved
work together to find the best way to move forward using
scientific methods.

An additional point to understand is that Kanban allows a business to change in an evolutionary way. There should never be a sudden shock to the system. This keeps the business running smoothly while improving continuously as you will see.

Chapter 2. The Kanban Board - Deciding on your Workflow

What is a Workflow?

A workflow is nothing more than a sequence of connected steps in producing something when we work. Every team, department and individual, has a set of steps that they take in order to work together to solve a problem, build a product or make a sale.

In the case of the kind of supermarkets that Toyota studied, the workflow can be (greatly) simplified down into three steps:

- sell product (gap appears on shop floor)

- replenish shop floor (gap filled on shop floor, gap appears in stock room)

✓ replenish stock room (supplier fills gap in stock room)

These three steps fulfilled for each product, keep the supermarket fully stocked.What is my Workflow?

As with the above, the workflow that you use will be strongly dependent on the work you or your organisation are doing. My software development team would work on development tasks each day. Their workflow was similar to the following:

- to do
- in process
- to verify

✓ done

I once worked in tele-sales, attempting to sell gym membership by giving away a free trial. My team had the following workflow:

- make call to promote free trial
- get lead (trial accepted)
- follow up on lead
- close deal

So in short, your workflow will be the set of steps that a given individual or group of individuals in your organisation use to solve problems, create a product or make sales.

The Kanban Board

A kanban board is nothing more than a highly effective means of visualising your kanban workflow. Although simple, it is very effective and definitely key, in focussing individuals within an organisation on using the kanban system.

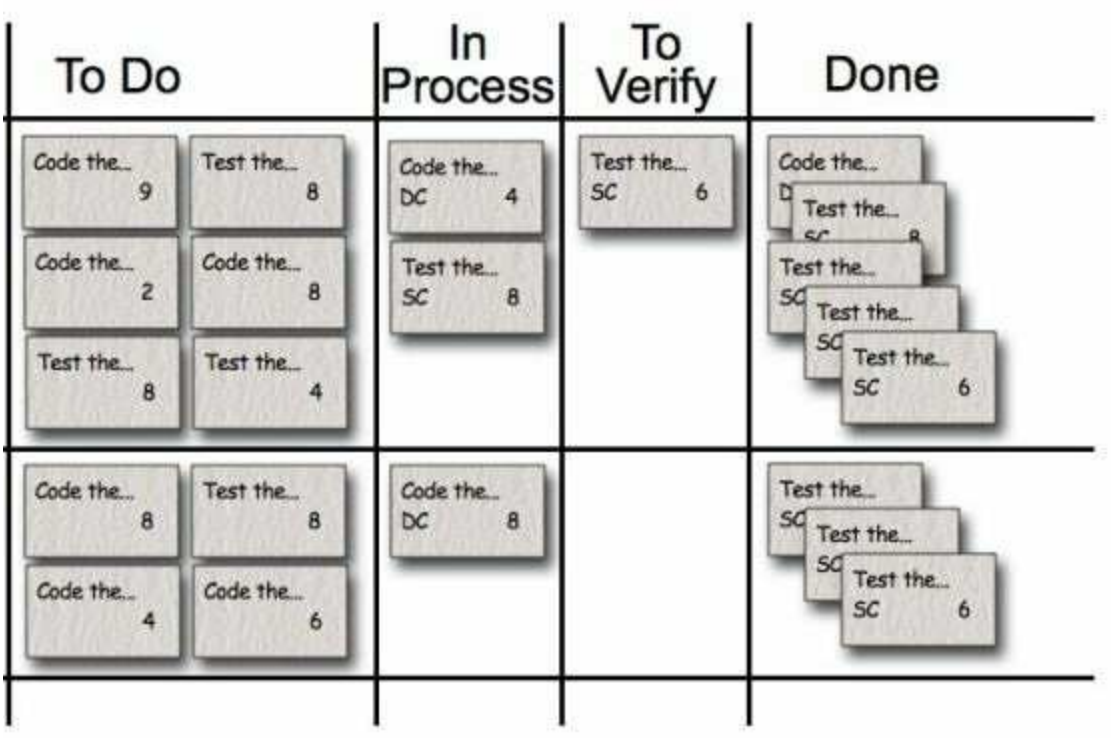


image adapted from xprogramming.com

The board essentially has a column for each step on the workflow and a row for each item of work. For example, in a software support team, the rows may correspond to issues such as:

    - add more memory to server
    - fix caching problem
- reinstall windows on Fred's PC

The columns may be:

    - to-do
    - in process
    - to verify
- done

The team or individuals involved can gather around the board to see their progress.

Each task on the board is written-up on a card or post-it note, that can be moved along from left to right to show progress along the workflow. The aim of the game is to reach the end of the workflow for each task as soon as possible.

Digital Boards

There are both digital and physical versions of kanban boards available dependent on your preference. Some prefer the "touchy-feely-ness" of a physical board. I personally have found much efficiency in using a digital board, such as storing acceptance criteria, related designs and schematics, manageability for disparate teams and historical searching of defects. This is a topic for another book, but within my teams, I have used Jira with the greenhopper plugin to great effect.

Chapter 3. Determining Priority

In any organisation with a multitude of issues and tasks, there is a need for a definite method of priority setting. Simply put, this ensures that the most important task will be carried out as soon as possible. This goes back to the old "80, 20" rule which states that 80 percent of the value in any organisation usually comes from 20 percent of the effort. However, this value can only be realised early if the correct 20 percent is started and completed as soon as possible. On a kanban board, the most important task cards are placed higher up the board to signify their priority. So how do people usually assign priority, using kanban? There are usually two methods widely used to decide

Method 1: Assign a priority maker

This relies on one key person. Usually it is a business manager, product owner or one experienced in the relevant product or service domain, making decisions on which tasks are most important and should be worked on first, as well as which order that they should be worked on. I have used this method in the development of software products, where a product owner or customer represents the business vision, liaises with stakeholders and is best placed to decide the priorities.

Method 2: Let the team decide

In a team oriented environment where a number of experts are well placed to make decisions on their domain, this method can allow a discussion around the board to work out the priority of tasks. This can be used in small hardware maintenance and support teams, particularly those with one customer.

Which method to choose

Each method can work. However, I have found that in each situation it benefits the team to have at least one person arranging the board and taking the calls as issues come in. Therefore, my personal preference is method one. I also found that the person arranging the board may as well be the one who sets priority, has a single view of the whole spectrum and can solely represent the customer.

Chapter 4. Understand The “Pull System”

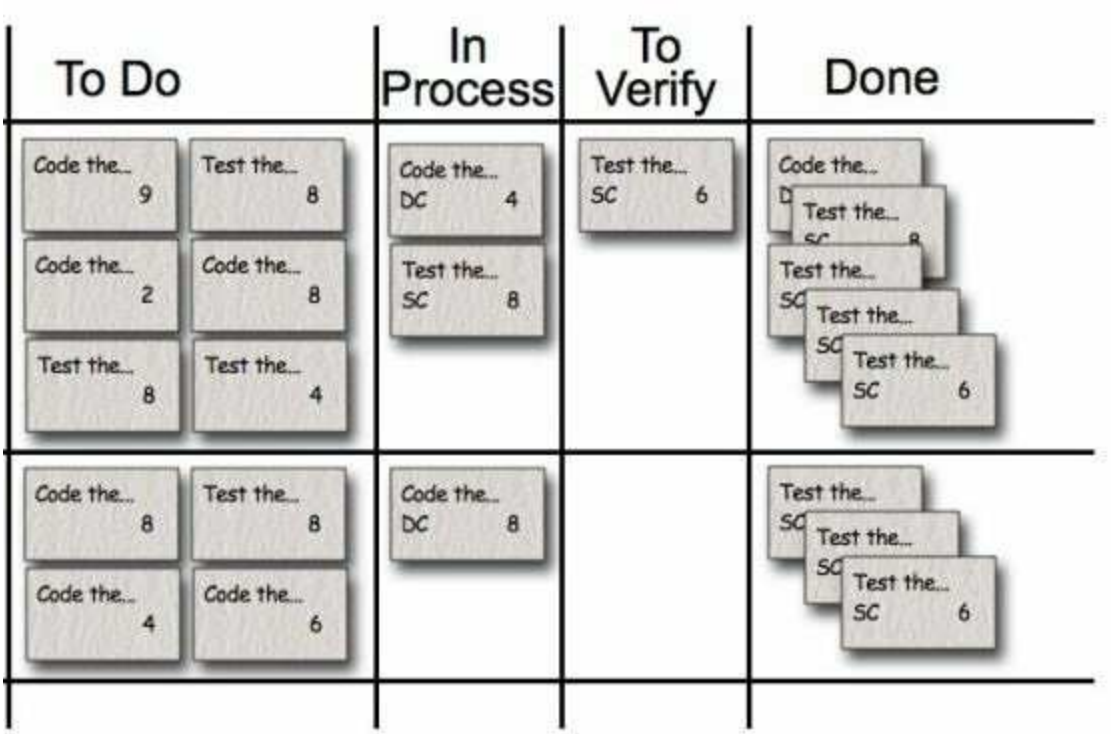


image adapted from xprogramming.com

Although there is no rocket science involved in it, one of the key concepts in Kanban is that of the “pull system”. Let us focus on our workflow and the rows of tasks on the board. If we imagine that we have only the columns:

- to-do
- in process

- to verify

✓ done

and then we have three roles of: product owner, hardware support engineer and QA tester.

The product owner discovers that her website is down and tells support. An issue then needs to be fixed. This results in a kanban card. As it is top priority for the owner, this task goes to the top of the board and is *pulled* into the "to-do" column. As soon as the hardware engineer has finished his tasks, he *pulls* the new task into the "in process" column. It is a quick fix and just needs a server to be restarted. When this task is complete, it does not get pushed into the QA tester's "to verify" column. The QA tester finishes what he is doing at that time and then *pulls* the task into the "to verify" column. When the tester has confirmed that the server is fully functional, the task is *pulled* into the "done" column. The QA tester then monitors the queue for more tasks.

The key thing in the example is that each team member will "pull" another task from the queue when they are free. In the same way as Toyota used this simple system to control the call for more stock, here we use the system used to control the call for more work.

Exercise - 1: Visualise your workflow

Workflow

Decide on the workflow most appropriate for one of the teams in your business (or for yourself if you don't have a team). This should be the set of steps necessary to get a set of tasks done specifically for the team doing the work!

Draw it as a set of steps such as:

Call customer -> Acquire Lead -> Close Deal

Kanban Board

Go to a white-board (if you don't have one use a flip chart or piece of paper for now). Set up the columns for your workflow steps and draw a few rows. This should create a grid to place your tasks in.

You can use 3x5inch cards or post-it notes to write your tasks.

Priorities

Who would assign priorities for getting work done in your team? Do you remember the usual choices and what would influence these choices? Write your answer down in your notes.

Pull system

Think about the different roles in your team(s). Who would pull work across the board? Write it down in your notes as these people will form the Kanban team.

Chapter 5. Establish Work In Progress Limits

The idea of Work In Progress (WIP) limits allows individuals and organisations applying kanban to manage the level of multi tasking allowed at any one time. In short, it decides the amount of work that should be in progress at any one time, for each step in the workflow.

Take for example the telesales workflow in my old gym:

- make call to promote free trial
- get lead (trial accepted)
- follow up on lead

✓ close deal

For any given sales consultant, the sales team will need to decide how many of each step they should allow to run concurrently before that consultant is overloaded. For example, it may be that any sales consultant should engage in no more than 10 calls in a day, concentrate on 6 leads, follow up on those 6 and aim to close 3 deals that day. Juggling any more than those numbers may be overload for the sales consultant and an inefficient use of her time. The idea is that focussing on a limited number of tasks will yield better results than being torn between too many.

Each type of business will need to establish the WIP limit for their workflow so that they can make sure they are operating at the optimum level of efficiency. This should be expected to vary from one business to another in accordance with the domain at hand.

Key Points - WIP Limits

The key thing here is that multi-tasking has often been seen as a good thing. While the ability to multi-task is a good thing and often necessary. The fact is that as human beings, we achieve much better results on our tasks if reduce the number of concurrent tasks we work on.

The idea of WIP Limits is to find the optimum number of tasks to maximise productivity whilst preventing overloading.

You can decide the rules here. It can be 1 task per person or it could be 3 tasks per person dependent on the task.

The only ask is to make sure that we don't set the limit so low that we are not as productive as possible or so high that the team become worn-out!

This can be done by looking at historical data for your team and working out what has worked or failed in the past and how many tasks they are delivering comfortably at present.

More on this later along with the metrics used to help work this out!

Exercise - 2: Set your WIP limit(s)

WIP Limit(s)

Think about the WIP Limits that are relevant to your team.
Do this in conjunction with your delivery time.

What is the average time needed to deliver your most typical tasks?

When do they typically need to be delivered (ie. once a week, once a month, once a day?)

What is the best approach to WIP Limits to ensure this is delivered consistently - i.e. do different roles need different WIP Limits?

Did you add some buffer - so if you need to deliver in 7 days can you complete the task in 5 days to give yourself some room?

Write down your WIP Limit(s) in your notes.

Chapter 6. Measure and Manage Flow

What is Flow

Flow is the smooth progression of work through a system.Excellent flow means a more productive team!" But how do we improve flow?

We do so by minimizing:

- hold-ups
- wastage
- bottle-necks

So in order to do this we:

1. gather data based on metrics.
2. look at our kanban board for bottle-necks.
3. problem solve to get work moving quickly.

Key Metrics

CYCLE TIME (CT)

The time an item takes to get from the To Do column to the Done column.We usually want Cycle Time to be as low as possible.

THROUGHPUT (TP)

This measures the amount of work done or the number of items completed in a given time frame.

LITTLE'S LAW

Little's Law states that TP = WIP/CT.

So:

- for any given Work In Progress
- throughput will increase
  - if cycle time can be reduced

Little's law is helpful to increase flow.

QUALITY

We should track quality over time. One example is:

- Number of defects OR
- Number of bugs OR
- Number of issues raised

We should aim to minimize these over time.

DUE DATE PERFORMANCE

For tasks with "due dates" we can track

- number of days past due date
- number of days under due date

Then we can solve problems to help meet or beat the due date

Exercise - 3: Measure Your Flow

Measuring Flow

Now we are going to see how well your team(s) areflowing.

Tasks on your board can be divided into different “classes”.
For example in the software world we have:

- defects
- stories
- coding tasks

In sales there may be:

- follow up calls
- courtesy calls
- etc.

Based on these classes for your business, you can decide on an acceptable cycle time with the organisation. Pick one class:

What is, the ideal average cycle time (usually a number of days or hours) required to deliver this class of tasks?

What is the actual average cycle time for this class of tasks?

Using Littles Law (Throughput = Work In Progress / Cycle Time) what is your throughput for each class?

By adjusting WIP you can affect throughput. Ideally we want a low Cycle Time to get a high throughput.

If anything does not make sense, go back to the previous lecture.

We will come back to how we use models to improve throughput and flow.

Chapter 7. Making Policies and Processes Explicit

Understand Toyota's Six Core Rules

Toyota developed six core rules within which kanban are operated. Being the ninth largest company in the world (by revenue), it is a pretty good example of how decent working practices lead to success. Let us explore their rules and how they apply to any business, team or individual.

1. Do not send defective products to the subsequent process

In other words, we must make sure that the work in the earlier workflow steps is done and complete before moving to the next step. If we become slack in any step, we create inefficiency. A problem with our server must be fully fixed before the QA engineer tests the fix, or her time will be wasted.

2. The subsequent process comes to withdraw only what is needed

Applied to our kanban board, this means that the later step in the workflow takes only the task that was completed in the previous step.

3. Produce only the exact quantity withdrawn by the subsequent process

The key here is that no more work is produced than what is necessary to complete the task in the previous workflow step. This is a vital rule in the efficiency plan.

4. Level the production

This is concerned with finding the constant rate at which your team or organisation can output their product or solve problems. This meets demand consistently, whether it goes up, or goes down.

5. Kanban is a means to fine tuning

In other words, simply by using the system, you will start to smooth out your processes as time goes on, again improving efficiency.

6. Stabilize and rationalize the process.

This means that whatever process you pick should eventually gain a consistent structure.

Chapter 8. Create and Refining your Process

As mentioned throughout this book, kanban can be very much adapted to your business, team or organisation. Therefore, creating a process that works for you is not something I can prescribe. I can however explain the process I have observed at global blue chip companies. Let us take the example of the last major media company I worked with. The team in question were a support team, tasked with supporting the hardware, web servers and deployments for the back end code that supplies data to iPad Apps. Their process was as follows:

Workflow

The team workflow consists of the following steps:

- *New Issue*
- *In Progress*
- *Blocked*
- *Tested*

- *Done*

Priority

Priority is set by the “Kanban Master”. The Kanban Master is a partly technical project manager who has an excellent relationship with the team and business. He balances the priorities of all the different product owners in the business.

Work In Progress limits

- No more than one task should be taken on by each person at each step in the workflow

- Deployments to the app server can be done in parallel as they are quick

Core rules

- No one goes directly to the team to ask them to work on tasks, they must go to through the “Kanban Master” to allow the team to do their work.

- Every day a stand-up meeting is held to co-ordinate the team. All must attend.

- A planning meeting is held once every two weeks to plan any long-term work. This means that the team are doing what is known as “scrum-ban” as this is a scrum planning meeting within the kanban process.

- Only the team can override the WIP limits

Refine

A retrospective meeting is held before every planning meeting. This is to make sure that the team continuously improve. This is in the same vein as scrum involves retrospectives.

Exercise - 4: Make Policies Explicit

Now we are going to make some explicit policies. These should always be agreed with your organisation. Key upstream and downstream stakeholders also need to be involved.

Upstream stakeholders have an interest in any inputs to the team (raw materials, instructions provided) and downstream stakeholders are interested in any outputs from the team (the final software, the final product the, final service)

Let's get it started.

Examples of policies were mentioned in the previous lectures. However all of the following are examples of policies.

- WIP Limits
- Desired Cycle Time for various class types (aka Lead Time)
- Desired Throughput for various class types
- Schedules for Prioritisation Meetings to be held for incoming tasks
- SLA (Service Level Agreements on delivery times)

What policies would you need to set to keep your Kanban team(s) running?

Would you use scrum-ban?

Which stakeholders would you need to involved?

This will form a set of policies to agree with your organisation

Chapter 9. Using Models to recognize improvement opportunities

How do we Improve?

In order to improve we use specific models continuously.

We then implement these models within our Kanban Teams.

David J. Anderson recommends 3 Models:

- ✓ The Theory of Constraints (study of bottlenecks)
- ✓ The Theory of Profound Knowledge
- ✓ The Lean Economic Model

These are all options but let us concentrate on one well known model. Let us understand The Theory of Constraints.

The Theory of Constraints (study of bottlenecks)

In a nutshell, this theory says that in an organisation there will always be a small number of constraints holding back your ability to deliver products or services.We could be held back by:

- people
- processes
- equipment
    - ✓ or policies

There are 5 steps to dealing with these constraints. We will now go through each one.

There are 5 steps to dealing with these constraints. We will now go through each one.

. IDENTIFY

Identify the source of the constraint/bottleneck:

- analyse your kanban board
- analyse metrics
    - discuss with team and management

2. EXPLOIT

Once you find the constraint, ensure you exploit max capacity to solving it eg.

- if constraint is machinery - is it running ALL day?
- if constraint is people - do they have ALL the resources they need?

etc.

3. SUBORDINATE

This means to de-prioritise other things so they can focus on the constraint. E.g.

- do policies need to change to allow people to spend more time on fixing defects?
- do people need to be re-prioritised to focus on fixing defects rather than starting new work?

4. ELEVATE

Here we further increase capacity if the exploit step has not worked. E.g.

- do we need new machinery to get moving?
- do we need new people or training for people?

5. REPEAT

This means to go back to step 1 and look for constraints again.

The likelihood is that constraints have cropped up somewhere else so stamp them out early!

These steps help increase flow. They also help to minimize cycle time, and minimizing cycle time is the name of the game.

Exercise - 5: Use Models To Recognise Opportunities for Improvement

Now we are going to think about how we can use models to improve. There are many models but I have shown you the theory of constraints (TOC):

Would you use the TOC model for your business?

Look at your Kanban board. Where do you see bottlenecks/constraints arising?

What kind of problems would cause these constraints? E.g.

- *people problems (eg. resistance to using Kanban)*
- *processes (eg. no process for dealing with complaints)*
- *equipment (eg. not enough PCs for all staff)*
- ✓ *policies (eg. too many meetings and no limit to them)*

How would you solve these problems in your business

I recommend using a retrospective meeting such as that I describe in the upcoming Scrum section to draw out any issues.

Chapter 10. Using Kanban with Scrum

Scrum-ban

What NOT to expect

As you know this is NOT a scrum or scrum-ban class, but I thought I would include some lessons on scrum and a few words on scrum-ban since scrum-ban is an emerging method and since scrum is often used in conjunction with kanban within departments in industry.

What is Scrum-ban

Scrum is a simple method for managing and completing complex projects.

Kanban is a simple method for just in time delivery of products or services.

Scrum-ban is a blend of Scrum and Kanban and it initially emerged as teams started to transition from Scrum to Kanban. However, I have used it rather to combine the pros of both methods.

I would say that this is a reasonably new, young method and as a scrum expert I would say that it has not been very well defined (you will see there are not a wealth of books on the matter). Let me give you a few pointers.

What are Key Differences between Kanban and Scrum?

Scrum	Kanban	
Iterations	No Iterations	In Scrum we have iterations of under a month called Sprints. There is no fixed period of time for delivery prescribed in Kanban. There are various target Cycle Times for different classes task.
Velocity	Flow and Cycle Time	Velocity measures the size the work completed in each iteration (Sprint.)
Scrum Master	No Scrum Master	The team own the process in Kanban. The process evolve and is not prescribed. Though I mention a Kanban Master in the class as a mad up person who helps suppor it. This is just a role we ma up because it suited us. It w not prescribed by Kanban.

When to use each method

This is down to your judgement, however, I have used Scrum-ban when I have a team that needs to accomplish both delivery of a complex long term project AND just-in-time maintenance or support of a product or service.

The best example is when you have a team that has both a fixed deadline for a project and an element of maintenance or support. Then you can use:

Kanban: for just in time maintenance, sales or support

Scrum: to deliver to the fixed deadline or scope in iterations

How I use Scrum-ban

I have used a blend of Scrum and Kanban as above in a very simple way:

- ✓ Keep the Kanban process as described in this course and use it for rapid fire in-coming support/maintenance/service oriented tasks
- ✓ Keep the Scrum framework as described in the upcoming lessons for any well defined project work with a fixed scope or deadline
- ✓ Add a planning meeting and a retrospective on an sprint by sprint basis (a sprint is an iteration - see the next lesson to understand sprints).
- ✓ Separate your board or have two boards (usually with the same workflow).
    - Scrum board for planned work
    - Kanban board for just in time work

Therefore the Scrum-ban method is simply the Kanban method with retrospective, planning and a separation of the board. It allows planned work and un-planned work to co-exist happily for the same team.

What is Agile Scrum?

In a nutshell, agile is an umbrella term for a set of frameworks and methods. These use iterative time boxed approachesand focus on building products incrementally from the beginning of a project rather than delivering it all at once at the end. Agile basically encourages frequent inspection and adaptation plus teamwork, accountability and self-organization. Moreover, the engineering behind agile projects allow speedy delivery of first-class products and their business approach aligns development with company goals and customer needs. Agile often works by breaking down projects into tiny user functionality bits called user stories, which are prioritized and then continuously delivered in short cycles (mostly 2 weeks) called iterations.

Scrum is an agile framework that traces back to early 90s, which provides a framework for managing complex product and software development with iterative and incremental processes. Scrum is a very popular framework for implementing agile project management. Although some people think that agile and scrum is the same thing, this is wrong. While there are many frameworks for implementing agile, scrum is the most outstanding owing to its specific concepts and practices categorized into roles, time boxes and artifacts. And although scrum is an agile technique that can be used for any project, it's mainly used for developing different software. It's mostly suitable for projects with requirements that change and develop very fast.

Agilemethods (including Scrum) implement the agile manifesto of:

- Individuals and interactions above tools and processes
- Working software over comprehensive documentation
- Customer collaboration above contract negotiation

✓ Responding to change over following a plan

Scrum Theory

Scrum is based on empiricism or empirical process control theory. The simple idea behind this theory consists of three principles: transparency, inspection and adaptation. Transparency (honesty) among the scrum team members is emphasized in that the product functionality is not 'done' until the team says so.

Transparency creates trust among the team members and enables them to consistently check up the progress (inspection) and make improvements based

on what they see (adaptation). These improvements include practices, communication, sticking to values, or otherwise. Consistently inspecting and adapting is quite powerful stuff. In so doing, they are improving over and over again, during and after the release of a product a feature that's not possible with other models of development.

Generally, the components of scrum framework are:

1. Three roles: Product owner, scrum master and scrum development team

2. Sprints: A project iteration of less than a calendar month

3. Scrum events/ceremonies: Sprint planning meeting (what and how meetings), daily scrum meetings, sprint review meeting and sprint retrospective meetings.

4. Scrum artifacts

Scrum has three artifacts:

a) Product Backlog: a prioritized backlog with end user requirements; the product owner is responsible for this (refer to the next section)

b) Sprint backlog; Elected items from the product backlog. It's like a mini-plan for achieving the sprint goal and delivering the product increment. Forecast the functionalities in the next increment and what needs to be done to deliver that functionality. The development team are responsible for this.

c) Product increments: At the end of each sprint, the new product increment should be in a functional state and meets the scrum team's definition of "Done". If all works well and the team's estimation stays on track, the increment has all sprint backlogs' items, meeting the team's definition of done.

To help you to understand this better, let's talk more about these components of the scrum framework:

1. The three roles

The entire scrum team is made up of three roles.

The product owner (PO)

Before carrying out any scrum project implementation activity, you need to find a relevant person who is willing to be a product owner.

The product owner is a person who guides the team towards creating the right product and is solely responsible for managing the product backlog by:

- ✓ Ordering the product backlog items inline with the goal and vision
- ✓ Ensuring these items are clearly defined and explained to all team members and that the backlog is visible and transparent enough to the team
- ✓ Giving the team their next agenda and optimizing the value of the team's work.

However, although all these can be done by the development team, the PO remains accountable. If there are changes to the backlog item's priority, the PO must be involved. For the project to succeed, the PO's decisions need to be respected.

Scrum master

TheScrum master is responsible for explaining and enacting scrum and makes sure those outside the team understand whether their interactions with the scrum team are helpful or not. The Scrum master serves the product owner, the team and the organization in the following ways:

- ✓ Helps PO find effective product backlog management techniques.
- ✓ Helps the scrum team understand why the product backlog must be clear and concise.
- ✓ Helps the development team develop high quality products
- ✓ Facilitates scrum events when needed
- ✓ Acts as a coach; teaches both the team members and organization how scrum operates
- ✓ Protects the team from external interruptions and gets rid of any impediments to ensure peak performance
- ✓ Plans the implementation process in accordance with the scrum theory

Scrum development Team

This team consists of professionals that work towards delivering an almost releasable increment of what's referred to as a "done product" after the end of

each sprint. They have every skill necessary to create what's referred to as "a product increment".

- ✓ They're solely responsible for creating any increment.

2. They decide on the number of priority items or user stories to complete during a sprint.

As a member of this team, you're referred to as a developer regardless of what you actually do (no titles). In addition, there are no sub-teams when it comes to addressing various issues like business analysis or testing; everything is done as a team. The size of the team must be small enough to stay agile and large enough to finish all the chosen tasks in a given sprint. Select a development team of three to nine people to avoid issues such as poor delivery of potentially releasable product increments (for extra small teams) and poor coordination or management (for extra large group). Unless the PO and scrum master are also executing sprint backlog's work, they're not include in this count.

Key Characteristics of the Scrum Team

-Self organization is important in all agile scrum projects. The team is self-organized in that there is no overall team leader or project manager to impose stuff on the team. Decision making includes all team members (all inclusive). In agile scrum, the PO and scrum master support the whole scrum team but the team decides on what and how much to do in a given sprint. The scrum team is also cross functional, which means everyone in the team is capable of and involved in taking any feature from idea to implementation without outside help.

-Continuous improvements; inspect & adapt, is another central aspect in scrum in that the scrum team must regularly inspect and assess their products and processes for adaptation and improvement purposes. Doing this optimizes results, increases predictability and therefore minimizing the overall risk. Scrum works on the maxim that requirements could change quickly or cannot be fully known at the start of the project.

-Communication is another cornerstone of scrum framework. To identify and prioritize functionality, the PO works closely with the scrum team. This functionality is presented in form of user stories and then stored in a product backlog. (I'll tell you more about user stories and product backlog later on).

2. The Scrum Events

Scrum projects progress through a series of sprints which form the backbone of these projects. A sprint is a project iteration that takes a period of time less than a calendar month, within which the development team works on their tasks until they develop a working increment of the product.

Sprint planning meeting: Time Box: 8 hours for a one month sprint

Planning Meetings are held at the beginning of each sprint which lasts a maximum of 8 hours (for 1 month sprints). It is attended by product owner, scrum master and scrum team.

During the sprint planning meeting, the PO submits and describes the prioritized product backlog and explains to the team about the sprint goal and product backlog's top items. The team agrees on which prioritized items to complete during the upcoming sprint and then shifts these items from the product backlog to the sprint backlog-a list of necessary tasks that they'll do during the sprint to attain the sprint goal. In doing so, they break each product backlog item into specific tasks they can effectively do during the sprint. Conceptually, the team selects the items to complete starting from the topmost, and then draws a line on the prioritized backlog after the lowest high-priority item they've selected.

After this meeting, the sprint kicks off. During the sprint, the development team takes a few ideas, create them and tests their functionality. In the end, these features become created, tested and eventually integrated into the developing product or system.

Daily scrum meeting; Time Box: 15 minutes

Every day, during the sprint, the scrum master, scrum team and or product owner (optional) must attend a daily scrum meeting of less than 15 minutes.

During these day to day scrum meetings, the team members talk about what they worked on the previous day, what they'll work on that day and identify any setbacks to progress. This daily scrums synchronize the scrum team's work plus their project knowledge, communication and decision making is enhanced.

Sprint review: Time Box: 4 hours for a 1 month sprint

At the end of each sprint, the scrum team carries out a sprint review which lasts for four hours for a 1 month sprint. During sprint review, the team demonstrates the new functionality to the PO scrum master, customers and other stakeholders/sponsors interested in providing feedback that may perhaps influence the following sprint. The relevant stakeholders express their impressions and clarify whether their user stories/requirements are well implemented. The PO ascertains whether the product is really "Done". This feedback loop could lead to changes to recently released functionality or just lead to revising or adding items to the product backlog.

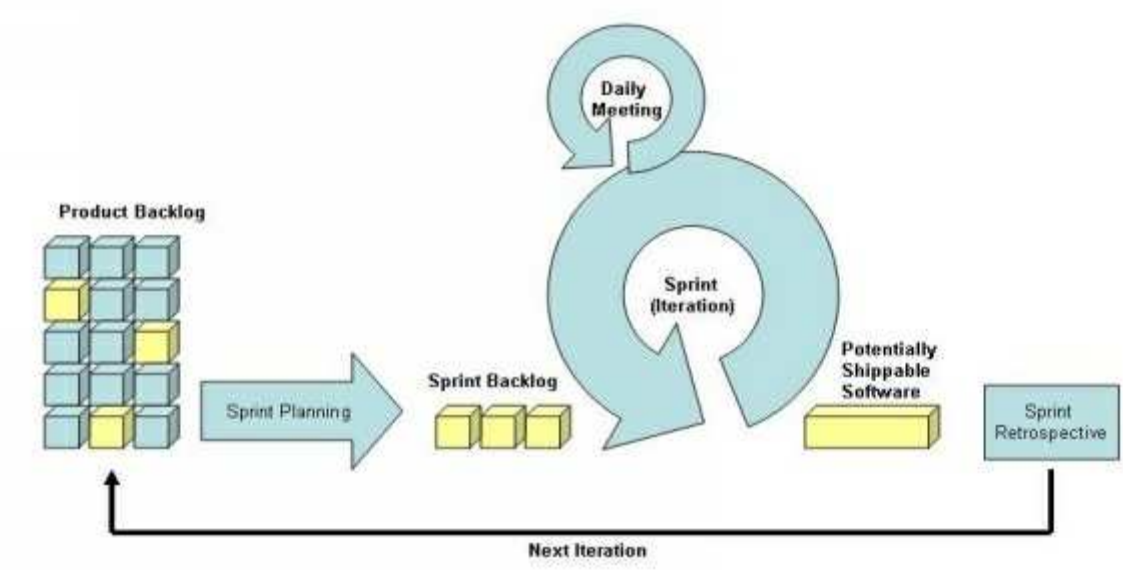
Sprint retrospective meeting; Time Box: 3 hours for a 1 month sprint

TheSprint retrospective meeting is another event in scrum attended by the entire scrum team at the end of each sprint. The entire team comes together to reflect on the sprint that has just ended as well as spot any improvement measures needed. They generally talk about how to make their work more effective and efficient in the future. While the sprint review is on the product, sprint retrospective meetings focus on the process.

3. The Scrum Skeleton

A very easy way I'll use to further explain the agile scrum process to you is using a scrum skeleton.

Here is diagram showing a scrum skeleton



On the left side of the skeleton, you can see the product backlog and a sprint backlog is adopted by the team, split into tasks, and worked on during a sprint.

The team carry out requirements gathering and specification update before the sprint, followed by design, implementation and testing.

The smaller circle you see above the large sprint circle represents the fact the team meets every day (daily scrum meeting) to inspect on progress and adapt their plan for that day. At the end of a sprint, they deliver a potentially shippable increment of the product. The business can then review the increment (sprint review) and if preferred, they can release the new feature(s).

The team engages in a transparent discussion about their progress during a sprint retrospective (inspect) thus improve (adapt) on stuff that needs to be improved or maintain the stuff that is going well. The cycle starts all over again and repeats until there is nothing more for product owner to add to the product.

The scrum skeleton shows that scrum is a simple and powerful mini factory, manufacturing shippable features in every sprint.

These agile scrum concepts and practices are applicable to any team or individual who needs to deliver a great project/product.

Exercise - 6: Consider Augmenting with Scrum

Now we are going to think about whether scrum is necessary for your business. Kanban is a different method from Scrum but the two are sometimes combined based on the different needs described earlier.

Would you use Scrum in your business?

Which elements of Scrum are most necessary for your organisation?

Would you use Scrum-ban in your business?

Remember to agree this as a policy in your organisation as with everything else.

Final Words

As stated in my other books, I recommend you trust your own instincts and intelligence, since I am sure you will agree that there is no substitute for good common sense.

CONGRATULATIONS, you now have a guide to using Kanban in your team or business.

Agile Product Management:

Product Backlog:

21 Tips To Capture and Manage Requirements with Scrum

Introduction

Thank you and congratulations on taking this class, "*Product Management: 21 tips to create and manage the Product Backlog*".

In this class, you will be given a multitude of proven tips to manage your product backlog as part of an agile scrum team.

I know you will get value from this class as it gives you a full introduction to the concept of the product backlog. I then walk you step by step through the steps involved in managing a backlog. Following this, I give you tips for improving product backlog management in your team or business from the ground up. Along the way, I give you plenty of examples and give you best practices for product backlog management within agile scrum. In this class, you will learn:

- A brief recap of agile and scrum principles
- What is a product backlog and how is it different from traditional requirements documents
- How to create a product backlog from a product vision
- What user stories are and how they are simpler for managing requirements
- Concise techniques for improving your product backlog management

So let ' s get started and let me teach you how to improve product backlog management.

Table of Contents

Agile Product Management:

Product Backlog:

21 Tips To Capture and Manage Requirements with Scrum

Introduction

Section 1: Introducing- Agile scrum

What is Agile Scrum?

Section 2: Introducing Product Backlog & User Stories

Product Backlog

Walkthrough: How To Create A Product Backlog From A Product Vision

Section 3: Product Backlog Management

Step 1: Convert All Ideas into User Stories

Step 2 - Prioritization

Step 3 -Estimate The Product Backlog

Step 4 - Refining and Grooming Your Product Backlog

Tips & Strategies

Tip #1: PO Should Know That He/She Will Be Expected To Present Enough Work That Can Last About Two Sprints Beyond The Current Sprint During All The Backlog Refinement Sessions For The Current Sprint

Tip #2: Avoid Scheduling Backlog Refinement During The First Or Last 20% Of The Sprint

Tip #3: The Backlog Items Must Be Fine Grained And Properly Understood By The PO For This Meeting To Flow Well. Define Initial Set Of Acceptance Tests Before The Meeting Occurs

Tip #4: Treat Every Backlog Refinement Meeting Just Like The Beginning Of Sprint Planning Meeting

Tip#5: Everyone Should Understand That Estimates Are Provisional Until A

User Story Is Accepted Into A Sprint

Tip #6: Strive To Optimize Your Time During The Meeting

Tip #7: Remember That Backlog Items Are Actually A Collaboration Between The PO And The Team

Tip #8: Keep Your Eye On The Goals Of The Meeting

Tip #9: Get Volunteers For Action Items For Any Unknowns Or Big Risks

Tip #10: You're Definitely Free To Split User Stories During This Meeting

Tip #11: Everyone Should Understand That Product Backlog Order Is Provisional Until A User Story Has Been Accepted Into A Sprint

Tip #12: Be Brave To Discuss A Couple Of Items Farther Down The Backlog

Tip #13: Let The Dev Team Review The User Stories To Be Discussed About 24 Hours Before The Refinement Session

Tip #14: Have A Mini Backlog Refinement Prior To Holding The Backlog Refinement For The Full Team

Tip #15: Track Progress Using Release Burndown Charts

Tip #16: Feel Free To Introduce Late Breaking User Stories. As You Do That, Strive To Minimize Them But Also Embrace Them As And When They Happen

Tip #17: Use One Issue Tracker and One Backlog!

Tip #18: Watch Out For Too Many Changes And Limit Your Scope

Tip #19: Ensure To Experiment With The Amount Of Refinement, Which Your Team Does

Tip #20: Ensure To Retrospect, Inspect And Then Adapt

Tip #21: Transparentize, Prioritize And Stick To The Product Vision

Conclusion

Preview of 'The Scrum Master Mega Pack'

Check Out My Other Books

Section 1:Introducing-Agile scrum

Before we can discuss the specific details about how to make the most use of a product backlog, let’s first understand what agile scrum is all about.

What is Agile Scrum?

In a nutshell, agile is an umbrella term for a set of frameworks and methods. These use iterative time boxed approachesand focus on building products incrementally from the beginning of a project rather than delivering it all at once at the end. Agile basically encourages frequent inspection and adaptation plus teamwork, accountability and self-organization. Moreover, the engineering behind agile projects allow speedy delivery of first-class products and their business approach aligns development with company goals and customer needs. Agile often works by breaking down projects into tiny user functionality bits called user stories, which are prioritized and then continuously delivered in short cycles (mostly 2 weeks) called iterations.

Scrum is an agile framework that traces back to early 90s, which provides a framework for managing complex product and software development with iterative and incremental processes. Scrum is a very popular framework for implementing agile project management. Although some people think that agile and scrum is the same thing, this is wrong. While there are many frameworks for implementing agile, scrum is the most outstanding owing to its specific concepts and practices categorized into roles, time boxes and artifacts. And although scrum is an agile technique that can be used for any project, it's mainly used for developing different software. It's mostly suitable for projects with requirements that change and develop very fast.

Agilemethods (including Scrum) implement the agile manifesto of:

- Individuals and interactions above tools and processes
- Working software over comprehensive documentation
- Customer collaboration above contract negotiation
- Responding to change over following a plan

Scrum Theory

Scrum is based on empiricism or empirical process control theory. The simple idea behind this theory consists of three principles: transparency, inspection and adaptation. Transparency (honesty) among the scrum team members is emphasized in that the product functionality is not 'done' until the team says so.

Transparency creates trust among the team members and enables them to consistently check up the progress (inspection) and make improvements based

on what they see (adaptation). These improvements include practices, communication, sticking to values, or otherwise. Consistently inspecting and adapting is quite powerful stuff. In so doing, they are improving over and over again, during and after the release of a product a feature that's not possible with other models of development.

Generally, the components of scrum framework are:

1. Three roles: Product owner, scrum master and scrum development team

2. Sprints: A project iteration of less than a calendar month

3. Scrum events/ceremonies: Sprint planning meeting (what and how meetings), daily scrum meetings, sprint review meeting and sprint retrospective meetings.

4. Scrum artifacts

Scrum has three artifacts:

a) Product Backlog: a prioritized backlog with end user requirements; the product owner is responsible for this (refer to the next section)

b) Sprint backlog; Elected items from the product backlog. It's like a mini-plan for achieving the sprint goal and delivering the product increment. Forecast the functionalities in the next increment and what needs to be done to deliver that functionality. The development team are responsible for this.

c) Product increments: At the end of each sprint, the new product increment should be in a functional state and meets the scrum team's definition of "Done". If all works well and the team's estimation stays on track, the increment has all sprint backlogs' items, meeting the team's definition of done.

To help you to understand this better, let's talk more about these components of the scrum framework:

1. The three roles

The entire scrum team is made up of three roles.

The product owner (PO)

Before carrying out any scrum project implementation activity, you need to find a relevant person who is willing to be a product owner.

The product owner is a person who guides the team towards creating the right product and is solely responsible for managing the product backlog by:

- ✓ Ordering the product backlog items inline with the goal and vision
- ✓ Ensuring these items are clearly defined and explained to all team members and that the backlog is visible and transparent enough to the team
- ✓ Giving the team their next agenda and optimizing the value of the team's work.

However, although all these can be done by the development team, the PO remains accountable. If there are changes to the backlog item's priority, the PO must be involved. For the project to succeed, the PO's decisions need to be respected.

Scrum master

TheScrum master is responsible for explaining and enacting scrum and makes sure those outside the team understand whether their interactions with the scrum team are helpful or not. The Scrum master serves the product owner, the team and the organization in the following ways:

- ✓ Helps PO find effective product backlog management techniques.
- ✓ Helps the scrum team understand why the product backlog must be clear and concise.
- ✓ Helps the development team develop high quality products
- ✓ Facilitates scrum events when needed
- ✓ Acts as a coach; teaches both the team members and organization how scrum operates
- ✓ Protects the team from external interruptions and gets rid of any impediments to ensure peak performance
- ✓ Plans the implementation process in accordance with the scrum theory

Scrum development Team

This team consists of professionals that work towards delivering an almost releasable increment of what's referred to as a "done product" after the end of

each sprint. They have every skill necessary to create what's referred to as "a product increment".

- ✓ They're solely responsible for creating any increment.
- ✓ They decide on the number of priority items or user stories to complete during a sprint.

As a member of this team, you're referred to as a developer regardless of what you actually do (no titles). In addition, there are no sub-teams when it comes to addressing various issues like business analysis or testing; everything is done as a team. The size of the team must be small enough to stay agile and large enough to finish all the chosen tasks in a given sprint. Select a development team of three to nine people to avoid issues such as poor delivery of potentially releasable product increments (for extra small teams) and poor coordination or management (for extra large group). Unless the PO and scrum master are also executing sprint backlog's work, they're not include in this count.

Key Characteristics of the Scrum Team

-Self organization is important in all agile scrum projects. The team is self-organized in that there is no overall team leader or project manager to impose stuff on the team. Decision making includes all team members (all inclusive). In agile scrum, the PO and scrum master support the whole scrum team but the team decides on what and how much to do in a given sprint. The scrum team is also cross functional, which means everyone in the team is capable of and involved in taking any feature from idea to implementation without outside help.

-Continuous improvements; inspect & adapt, is another central aspect in scrum in that the scrum team must regularly inspect and assess their products and processes for adaptation and improvement purposes. Doing this optimizes results, increases predictability and therefore minimizing the overall risk. Scrum works on the maxim that requirements could change quickly or cannot be fully known at the start of the project.

-Communication is another cornerstone of scrum framework. To identify and prioritize functionality, the PO works closely with the scrum team. This functionality is presented in form of user stories and then stored in a product backlog. (I'll tell you more about user stories and product backlog later on).

2. The Scrum Events

Scrum projects progress through a series of sprints which form the backbone of these projects. A sprint is a project iteration that takes a period of time less than a calendar month, within which the development team works on their tasks until they develop a working increment of the product.

Sprint planning meeting: Time Box: 8 hours for a one month sprint

Planning Meetings are held at the beginning of each sprint which lasts a maximum of 8 hours (for 1 month sprints). It is attended by product owner, scrum master and scrum team.

During the sprint planning meeting, the PO submits and describes the prioritized product backlog and explains to the team about the sprint goal and product backlog's top items. The team agrees on which prioritized items to complete during the upcoming sprint and then shifts these items from the product backlog to the sprint backlog-a list of necessary tasks that they'll do during the sprint to attain the sprint goal. In doing so, they break each product backlog item into specific tasks they can effectively do during the sprint. Conceptually, the team selects the items to complete starting from the topmost, and then draws a line on the prioritized backlog after the lowest high-priority item they've selected.

After this meeting, the sprint kicks off. During the sprint, the development team takes a few ideas, create them and tests their functionality. In the end, these features become created, tested and eventually integrated into the developing product or system.

Daily scrum meeting; Time Box: 15 minutes

Every day, during the sprint, the scrum master, scrum team and or product owner (optional) must attend a daily scrum meeting of less than 15 minutes.

During these day to day scrum meetings, the team members talk about what they worked on the previous day, what they'll work on that day and identify any setbacks to progress. This daily scrums synchronize the scrum team's work plus their project knowledge, communication and decision making is enhanced.

Sprint review: Time Box: 4 hours for a 1 month sprint

At the end of each sprint, the scrum team carries out a sprint review which lasts for four hours for a 1 month sprint. During sprint review, the team demonstrates the new functionality to the PO scrum master, customers and other stakeholders/sponsors interested in providing feedback that may perhaps influence the following sprint. The relevant stakeholders express their impressions and clarify whether their user stories/requirements are well implemented. The PO ascertains whether the product is really "Done". This feedback loop could lead to changes to recently released functionality or just lead to revising or adding items to the product backlog.

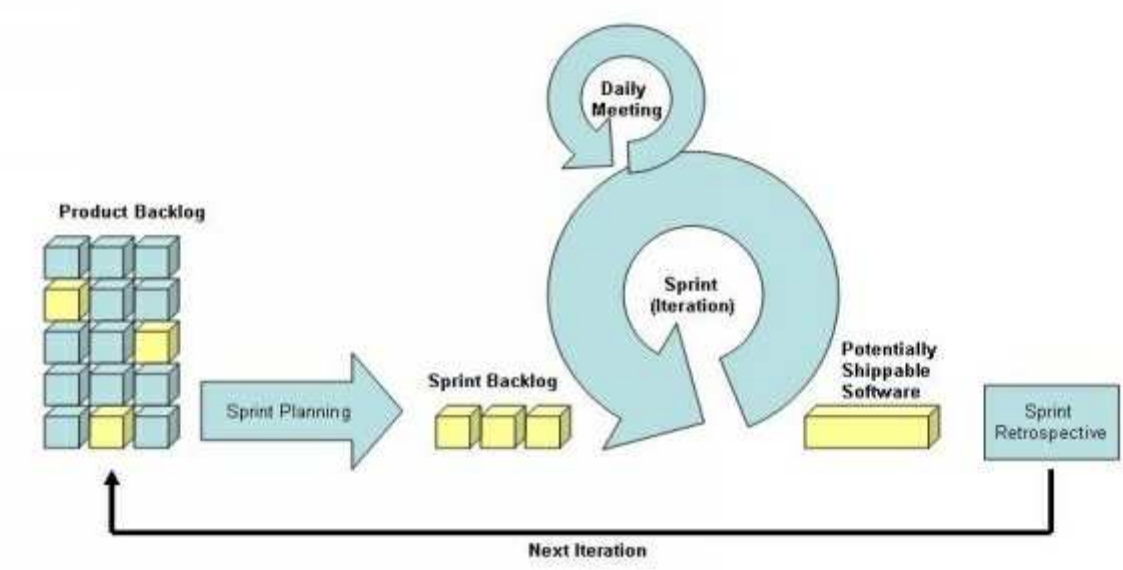
Sprint retrospective meeting; Time Box: 3 hours for a 1 month sprint

TheSprint retrospective meeting is another event in scrum attended by the entire scrum team at the end of each sprint. The entire team comes together to reflect on the sprint that has just ended as well as spot any improvement measures needed. They generally talk about how to make their work more effective and efficient in the future. While the sprint review is on the product, sprint retrospective meetings focus on the process.

3. The Scrum Skeleton

A very easy way I'll use to further explain the agile scrum process to you is using a scrum skeleton.

Here is diagram showing a scrum skeleton



On the left side of the skeleton, you can see the product backlog and a sprint backlog is adopted by the team, split into tasks, and worked on during a sprint.

The team carry out requirements gathering and specification update before the sprint, followed by design, implementation and testing.

The smaller circle you see above the large sprint circle represents the fact the team meets every day (daily scrum meeting) to inspect on progress and adapt their plan for that day. At the end of a sprint, they deliver a potentially shippable increment of the product. The business can then review the increment (sprint review) and if preferred, they can release the new feature(s).

The team engages in a transparent discussion about their progress during a sprint retrospective (inspect) thus improve (adapt) on stuff that needs to be improved or maintain the stuff that is going well. The cycle starts all over again and repeats until there is nothing more for product owner to add to the product.

The scrum skeleton shows that scrum is a simple and powerful mini factory, manufacturing shippable features in every sprint.

These agile scrum concepts and practices are applicable to any team or individual who needs to deliver a great project/product.

Section 2: Introducing Product Backlog & User Stories

Product Backlog

In agile scrum, a product backlog is a prioritized list of features that has short descriptions on all the various functionalities desired in a product.

A backlog is like a front door to the project since all the work the team will do comes from it. Unlike a requirements document, with scrum, there is no need to struggle with producing a lengthy product backlog at the beginning of the project and then leave it on your shelves to collect dust. Instead, at scrum we make a living product backlog, which must evolve, expand and contract following your project realities. Some backlog items can become irrelevant and must be removed while others need to be chunked into smaller stories. Basically, the PO, scrum master and scrum team brainstorms on the work needed to finish the product/project and then puts it on the product backlog, in priority order.

This initial product backlog can be used for the first sprint but the team continues to make the necessary changes to it as they learn more and more about the product and its customers.

Generally, a scrum product backlog contains anything relating to the product:

- ✓ Features (New features or product extensions to address new opportunities or markets, or features requested by customers or internal stakeholders)
- ✓ Technical stuff (engineering stuff, enhancements, UI and usability changes or improvements)
- ✓ Bugs (that need fixing)
- ✓ Knowledge acquisition (research, risks)

Product features/attributes comprises two types of requirements:

1) Nonfunctional requirements: they include requirements such as robustness, performance and usability.

2) Functional requirements; describe specific product features or functions such as sending an email or making a call.

The product attributes helps the team come up with a set of all possible solutions. Coming up with detailed balanced product attributes is a joint effort. As such, everyone should participate in the process. This is especially so because under-specifying attributes leads to a lack of guidance and direction while over-specifying product attributes leads to making decisions way too early and impeding the team's creativity. Features can be described using; use case, personas and scenarios, and user stories (which i recommend).

Ideally, the items on a product backlog should be expressed with regard to the value it gives to the user (customer or business). Functional requirements are expressed as features while non functional requirements are put, for example, like this "The product needs to be faster by x minutes" or "We need to ensure the product is more secure in the following manor".

All Generic operational qualities can often be captured as what is known asconstraint stories; visually describe the product or user interface design, for example, in form of mock-ups, sketches and screen shots. These aren't features as such, but, they're valid product backlog items.

The best way you can express features on the agile product backlog is in form of user stories. An example of a user stories would be "As a shopper, I want to review the items in my shopping cart before checking out so that I can see what I've already chosen."(More about user stories later).

The fact that the bug describes something different a user wants, similar to a new feature, it's also included in the scrum backlog. Technical work and research activities are also placed in the scrum backlog. For instance, technical work would be "upgrade all developers' working stations to windows". As for Knowledge acquisition, it could appear in the product backlog as an item about researching different types of JavaScript libraries and choosing the best.

Factors that influence product owner's prioritization include:

- ✓ Customer priority
- ✓ Relative implementation difficulty
- ✓ Urgency of getting feedback
- ✓ Symbiotic relationships between work items (e.g. B is easier if

we do A first)

To keep the product backlog in shape as the project rolls, the PO does the following activities:

-Adding and Prioritizing the Newly Added Stories

-Requesting the team to estimate new stories and re-estimate old ones to make them better understood

-Reviewing upcoming user stories and collaborating with the team in breaking the new items down

-Meeting with stakeholders and customers to review and add requirements.

Note that, the stakeholders and the team can create and manage the product backlog but the ultimate responsibility of prioritizing it lies with the PO.

What Makes a Good Backlog?

Characteristics of a good product backlog includes:

- ✓ It's prioritized to ensure the team builds key features first
- ✓ Has all the work necessary to complete the project
- ✓ It's a living document that constantly evolves to reflect the current project realities
- ✓ It's estimated by the team to help the PO gain clarity and ability to answer relevant questions like, "When will the stories be done?"
- ✓ It's DEEP

DEEP rule

As a summary to key attributes of a good backlog, the acronyms represents:

Detailed appropriately: users stories for the upcoming sprint must be well detailed and understood. Give fewer details for those that aren't going to be developed for awhile.

Estimated: Focus on correctly estimating the first priority items. Items further down the backlog can have less precise estimates since they're not yet well understood.

Emergent: The backlog changes over time (not static); user stories are added,

reprioritized or removed.

Prioritized: The most valuable items should appear at the top while the least valuable at the bottom of the product backlog. Working in priority order enables the team to maximize their product/system value.

Although having a good product backlog doesn't automatically guarantee a good end product, the lack of a good one often results in incomplete end products that do not meet the customer and stakeholder requirements. Moreover, creating a product backlog helps the entire team clearly picture what's left to be done and therefore avoid last-minute surprises.

User Stories

Agile scrum includes user stories, which stop us from focusing on requirements to discussing them. User stories are short descriptions of features from a user or customer perspective, written in form of a sentence or two showing a series of talks concerning the desired functionality. User stories explain what the product should do (end user desired functionality), but, not the details about implementation (How-Tos of building a product meeting those requirements).

User stories aren't just small text snippets; each of them is composed of three aspects

- ✓ Written description of each story for planning and recalling purposes.
- ✓ Conversation on the story to bring out its details.
- ✓ Tests that express them and recorded details to help determine when the story is done.

Each user story normally follows the format:

"As a <type of user>, I want><functionality> so that <some reason>."

User stories can be written by anyone (stakeholder or team) but the PO organizes each of the user stories into a list of backlog items. They're normally written on sticky notes or index cards, stored in a small box and arranged on tables or walls to facilitate discussion and planning. As such, every one embarks on talking about the features instead of compiling them.

Why user stories?

User stories are simpler for managing requirement and ideas and the best type of product backlog items due of the following advantages:

-One advantage of agile user stories is that we can alter their level of details- we can either write them to cover lots of functionalities or less. It's common to find large user stories known as epics. An example of an epic user story from a desktop back up product; "As a user I can back up my entire hard drive".

Evidently, epic stories are basically too large for any agile team to complete within one sprint/iteration. Therefore, this big user story is split into multiple smaller user stories before the team works on it. It can be split into dozens or hundreds- for example;

"As a power user, I can specify folders or files to back up on the basis of file size, date created and date modified"

To add details you can either:

-Split the user story into many smaller user stories or

-Add "conditions of satisfaction".

By splitting a large story into lots of agile user stories, one may naturally think that details have been added,after all, you've written more.

"Conditions of satisfaction" is just an advanced acceptance test (acceptance criterion) that will be valid once the agile story is complete. Think about the following agile user story case:

"As a marketing vice president, I want to choose a holiday season for reviewing the previous advertising campaigns, so that I can identify profitable ones"

To add details to the example above, you can attach the "conditions of satisfaction" as follows:

-Check that it works with major retail holidays: president's day, father's day mothers day, Christmas.

-Holiday seasons can follow from one holiday to the next (for instance Thanksgiving to Christmas).

Another advantage is that user stories stress on verbal communication. The imprecision in most written languages leaves no guarantee that a developer and a customer will interpret a statement similarly.

Moreover, user stories can be used for planning straight away. User stories are easy to estimate in terms of time and resources needed to develop;And, since user stories are short, they're implemented within relatively short project iterations.

User stories spare the details at first. You can write them then add the details once it's important to do so-this makes them perfect for time-contained projects. As a team, you can jot down something real quick to give you a general feel for the system, then plunge into details on a couple of the stories and start creating much sooner than a team feeling compelled to complete a whole requirements document first.

Principles for writing user stories

A key factor in agile scrum is writing good user stories that follow these key rules:

The Three C's

Good user stories are above mere statements. A good user story has three elements usually known as three C's:

1. Card

A user story should fit on 3"×5" note card capturing key info (not all the requirement details). This 'C' can refer either to the card or the optimal size of a user story. This written text is taken as an invitation to conversation. This card is presented in scrum meetings . The concept behind this card acknowledges that the team and the customer will discover the desired underlying system while working on it. And this discovery will happen during conversations and collaborations about user stories.

Usually, the card follows the format below;

As a user<user role> of the product, I can <action> so that< benefit>

In other words, a story's written text has to address the 'who, what and why' of the story.

2. Conversation

The PO facilitates a collaborative conversation that involves the team and all stakeholders. It's mostly an in-person conversation that has the real value of the story. The written card must be adjusted to reflect the present mutual understanding. The conversation is generally verbal, but, often has documents to support it and ideally different automated test such as acceptance tests.

3. Confirmation

Before a story is considered 'done', the PO must confirm that it's complete. The PO and the team check each story's 'doneness' considering the team's present definition of '"done"

A Specific acceptance criteria-different from their current definition of 'done'-can be created for special stories. However, the team should understand and agree upon the current criteria. All relevant acceptance tests must be passed.

INVEST

The acronym INVEST is used to verify that everyone understands the story and that the team can begin working on it. Here is what each letter stands for:

I-Independent

The team can implement the solution independent of other stories. The team must split technical dependencies as much as possible. This requires creativity, problem solving skills and agile technical practices (e.g. refactoring).

N-Negotiable

The scope of work must be flexible and not rigid like the traditional requirement specification. Moreover, story solutions aren't restricted but open for discussion and collaboration with development team responsible for finally deciding about technical implementation.

V-Valuable

The story's business' value (the 'why') must be clear to everyone. The "why" doesn't have to be from a user's perspective; it can address a customer's business need without giving a direct valuable outcome to the end user.

All stories should associate with clear business goals. However, a story doesn't have to be a solo marketable feature.

E-Estimable

The team needs to understand the story adequately to estimate the work's complexity and the effort needed for delivering the story as a potentially shippable functionality increment. Nevertheless, the team doesn't need to understand all the implementation details to be able to estimate the user story.

S-Small

The item has to be small enough to make it possible for the team to deliver a potentially shippable functionality increment within a particular sprint. In fact, every item on a product backlog should be made smaller as it gets close to the top- that's why the scrum team needs to do backlog refinement.

T-Testable

The entire team has to understand and agree on how to verify the story's completion. One way of determining this is by defining "Done". If everyone concurs that it's possible to implement the story in a manner that meets their current definition of "Done" (including the acceptance tests) within a single sprint, then the story is testable.

You can apply the INVEST criteria for any product backlog item including the ones not written as user stories.

Note: You can derive your initial product backlog from a product road mapand/or a vision board. In this class we will concentrate on the latter.

Now that you know some basics about product backlogs and user stories, we're going learn about the product vision and how to use it to create a product backlog in our next topic.

Walkthrough: How To Create A Product Backlog From A Product Vision

Certainly, working on a scrum project demands the entire scrum team to be on same page with regard to the desired finished product. And that's why every scrum team needs a product vision serving as their compass. The product owner, scrum master, development team, customers and other stakeholders must share this overarching goal.

Before I talk about how to create a product vision and most importantly how you can use it to create a detailed product backlog, let's first talk about some basic things:

What Is A Product Vision?

A vision clarifies why you're undertaking the project and the desired end state. Jonathan swift, an English writer once said that, a vision is the art of seeing invisible things. The product vision paints the future picture that draws people in. It describes who the customers are, what they need and how their entire needs will be met. A vision basically captures the product essence-critical info that the entire scrum team must familiarize with to develop and launch a winning product. Moreover, it helps in capturing the opportunities and the possible threats you face.

Writing and sharing a strategic product vision must put your product end user first. A powerful product vision should clearly describe your end consumers, their needs and your go-to-market plan. A product vision helps you come up with strategic goals and initiatives.

To create an effective vision, you must answer the five questions below:

1. Who are the target customers?

2. What essential needs does the product address?

3. What crucial product attributes really satisfy the needs identified?

4. What makes the product stand out from the existing products in the market, both from competitors and within the company?

5. What is the project timeframe and budget, for both developing and launching

the product?

Answering these five questions provides you the information for creating a business case and allows you to decide if and how to proceed with the project. Your vision is like your north star when it comes to deciding whether or not to develop the product; if adding a feature isn't helping the product achieve its vision then the feature should be discarded.

Creating the Product Vision

Vision-creating activities should be spearheaded by the product owner (who is directly responsible for the success of the product and its returns) but, he/she should collaborate with others. For innovative projects these people includes technical and business people such as product and user interface designers, marketers and developers. Creating a vision for highly innovative and complex products requires more effort. Activities such as market research and prototyping usually accompany major product updates and new product development. Compiling the relevant information in this case may take several weeks or even months and so, the best way for the team to carry out the necessary work is by running one or more sprint. On the contrary, creating the vision for a maintenance release or a small product update may only take a few hours or days.

The core of the product vision is describing the chosen customer needs and the key product features meeting those needs.

A product vision should appeal to our emotions and intellect. It should inspire and motivate people. Also, it should be stable and clear, short and sweet, broad and engaging. Generally, this means that the product vision:

- ✓ Should be easy to understand to avoid misinterpretation
- ✓ Adjustment should not change the product value propositions to avoid de-motivation, confusion and project failure
- ✓ The goal it describes should be broad and engaging enough to allow creativity as well as guide the development effort
- ✓ And finally, the information in the product vision should be critical to the product's success.

Before Creating a backlog from a product vision

Even with a clear idea for our new product or new features, it's not easy to

know which user stories to include in the backlog. Or even to identify who should be at the sprint review meeting to help us know whether we're developing the right product. In the worst-case scenario, we may implement the wrong stories, get responses from the wrong people and eventually make a product which no one wants or needs. That's why it's important to know how to make a backlog using the product vision.

To be able to create a product backlog from a vision, you need to ask yourself the following four questions.

Four questions

Before stocking the product backlog, we have established the need to answer the following four questions:

1. Who are the customers and users?

2. Why would they buy and use the product?

3. What makes the product unique? What are its key features?

4. What business goals should the product deliver and how are they met?

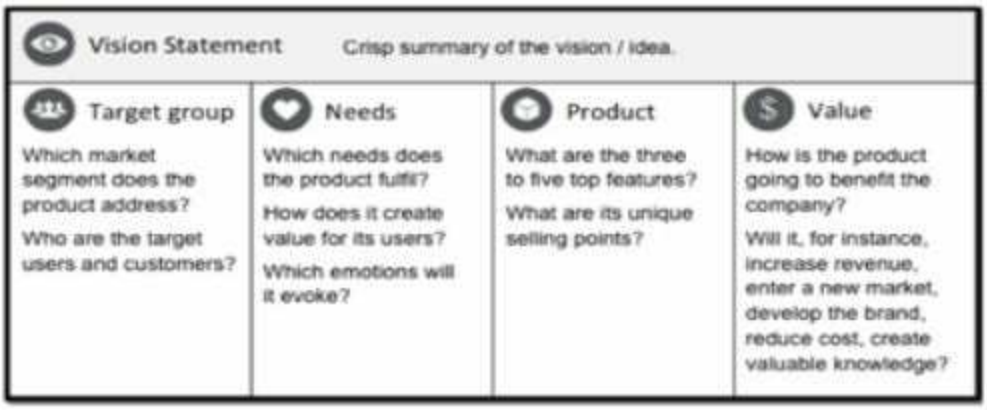
Writing meaningful user stories is impossible without knowing who the clients are and why they should use the product. Also, it's not clear what makes the product stand out or special, making informed decisions concerning the product functionality, the user interaction, the UI design right, and key technology and architecture choices. And, if we don't make the business goals and the product's business model clear, it may be hard to justify the project and attract resources.

When using vision to create a product backlog, the first thing you can do is create a vision board.

What's a vision board?

The vision board is a tool that captures important assumptions that impact the product success and allows the team to start the testing and refining their ideas.

This simple yet effective tool helps agile teams capture their vision and product strategy. In addition, it thoroughly answers the four questions above. The following picture illustrates the Vision Board, and its sections are explained below.


Vision Statement
Crisp summary of the vision / idea.
Target group
Which market segment does the product address?
Who are the target users and customers?
Needs
Which needs does the product fulfil?
How does it create value for its users?
Which emotions will it evoke?
Product
What are the three to five top features?
What are its unique selling points?
Value
How is the product going to benefit the company?
Will it, for instance, increase revenue, enter a new market, develop the brand, reduce cost, create valuable knowledge?

credit to: www.romanpichler.com

Vision Statement is a brief summary of your idea that explains your intention and motivation. It's best to limit it to two sentences.

Target Group describes the market segment or market you want to address. Clearly state who the product users and customers are likely to be.

Needs expresses the value proposition of the product: the pain and problems points the product gets rid of, and the gains or benefits it creates for the customers and the users. In this section, you should make it clear why people should use and buy your product and its value proposition.

Product gives a summary of the three to five features that make your product stand out and determine its success. This one should mainly correlate your product's unique selling proposition, and address identified needs. Make sure you stick to the five features limit and save the rest of the details for your backlog.

Value describes why investing in the product is a worthwhile venture and how to achieve the business goal. This section shows the possible benefits for your company such as enter a new market, increase revenue, reduce cost, gain a technological advantage or develop the brand.

To make this more concrete, let's have a look at the following example:

A Sample Vision Board

Let's assume that what we want is to create a new website for sports fans. The Vision Board could look like this in the end:

Vision Statement Develop a world class sports website that will benefit sports fans by giving them up-to-date news and results for the most popular sports in the world.
Target Group:
Sports fans of all ages across the world. Particularly for the top sports in the world (found by research).
Needs:
Feel fully informed on the main news headlines related to the top sports.
Feel fully up to date on the results related to the top sports.
Product:
Website with:
• news headlines from all sports
• sport specific headlines for the top sports
• sports results for the top sports
• editorially controlled news stories
Value:
Increased brand awareness and future opportunities for advertising revenue.

The overarching vision is captured on the board above. It clearly states that sports fans of all ages are the intended website users, particularly for the top sports in the world. It provides three explanations about why the sports fans would like to use the website, the key product features, and the business goal and how the business model might perform in the future. As you may have noticed, the info on my vision board is concise and coarse-grained.

On one hand, we're able to understand the new site's market, value proposition and its business model using the Vision Board above. On the other hand, due to its many untested assumptions, using the board to develop the product backlog would be quite a risk. What we need is to point out and validate its assumptions systematically.

Validation and research of the Vision Board

Apart from being an assessment and communication tool, the vision board gives you room to test your assumptions and capture any new insights you gain. When it comes to validating the Vision Board, it's best to first identify the utmost risk or the major uncertainty on the board. This helps in creating focus and implements a fail-fast: you quickly figure out what works and what doesn't as well as true and false assumptions. When working on your product, your greatest risk can be to misunderstand the user needs and potentially build a

product that doesn't give much value. Before you explore further about what features the product provides or how to implement it, you first have to test user need assumptions. You need to start with identifying the key assumption, the assumption that you must test now to avoid disappointments in the eleventh-hour – discovering very late that it's not worthwhile to solve the problem. After identifying the key assumptions, choose the best method for addressing it and those who should sit in your test group.

Test your Vision Board assumptions using methods such as direct observation – where you observe intended users and customers doing their business, for instance, observe how people are using sports websites on the multiple devices nowadays. And, problem interviews – you talk to members of the target group concerning their current experiences, for instance, what's good and bad regarding websites, mainly the ones related to sports, to understand individual goals and problems without referring to the solution. Doing the interviews and making direct observations will help you understand the target group better and assess how much value the product would provide. It can also help you update the vision board to reflect your latest thinking.

These measures will help you understand your target group more and assess the value your product will provide. And, you can update and change your board to reflect your current ideas. Don't be afraid of failure; early failure helps save your time and money.

After testing assumptions, you should complete the relevant data collection then analyze it. You can make radical (pivot) or incremental changes on the board using your new insights. For example, change incrementally by either, replacing a key feature or adjusting a need or radically making a change that affects one or more sections of the board. The path changes but the vision still stands.

From vision board to the product backlog

After validating the critical assumptions and the key risks on the board, use it for creating the first product backlog and identify the right stories, UI design and interactions.

When you have a refinedproduct backlog and a scrum team, you're ready to commence sprinting. Before we get to that, here is a quick summary of what we've learnt so far:

In Summary

- ✓ Before you write user stories, produce interface ideas, or user interaction (UI), ensure you know who the users and customers are.
- ✓ Discover the product uniqueness, your business goals and how you can achieve them.
- ✓ Create a product vision that prioritizes on the end users.
- ✓ Create a compelling product vision board that has a clear vision statement and that addresses all the possible assumptions.
- ✓ Validate your vision board using techniques such as direct observation and problem interviews. Follow the approach of identifying your biggest risk and attacking it first when working on a vision board.
- ✓ Create a backlog using the tested and approved assumptions only.

Section 3: Product Backlog Management

In addition to rendering release and iteration planning futile, a disorganized product hampers your ability to harmonize your product with the overall plan and destroys your credibility. In fact, there are so many misconceptions when it comes to managing scrum product backlog and this can create a huge predicament.

A well managed backlog boosts the chances of customer satisfaction and success. However, a poorly managed backlog could produce the wrong system, regardless of the team's top-notch skills. On top of the INVEST, Three Cs and DEEP rules on how your user stories product backlog should look, there are several tangible ways to deal with your product backlog. Before I give you some invaluable tips on how to manage and refine your backlog let's look at how to manage it step by step.

Step by step guide on how to manage the product backlog:

Step 1: Convert All Ideas into User Stories

At the start of any agile scrum project, having a nice list of a clearly defined product backlog items is very unlikely. Instead, you possibly have a pile of story note cards and perhaps one or two functional specifications. As previously mentioned, large user stories that cannot fit inside of a sprint are called epics. Your first step in managing the backlog is to convert all the ideas and notes into epics. As a product owner, your job is getting this sorted out and bringing some sort of order to enable the team to start estimating the backlog. Remember that we suggest they are written in the format we talked about.Turning the requirements into epics is often done with the team in a product backlog refinement meeting (all though the backlog takes a while to refine).

Similar to the product backlog, each user story evolves after a while. Your team is free to make changes to user stories by either adding information to them, breaking them down into smaller stories, or deleting them altogether. The ones that are presented during sprints are implemented and then delivered to the customers.

After converting all your ideas and notes into epics, you can proceed to estimating and prioritizing them.

A sample of a sports website product backlog of Epics (enlarge to view clearly)

Note: The User story title and acceptance criteria would usually be written on the front and back respectively of paper story cards. I have written them in columns to help visualize them in this class.

Epic	STORY DETAIL PAGE: As a Sports Fan, I want to see a 'story detail' page, so that I can stay informed with a detailed news story	ACCEPTANCE CRITERIA The user should see: - a story title - the amount of time since the news story was published in hours - a large headline image - a brief story summary - a detailed story consisting of paragraphs of text - related stories small icon - related stories small headline All stories should update dynamically without reloading the page The CMS should enable an editor to: - Supply relevant story metadata (as above) - Set related stories with the data above EDGE CASES:	
Epic	VIDEO: As a Sports Fan, I want to see video and audio related to the current story	ACCEPTANCE CRITERIA A user should be able to click and watch a video when it is editorially assigned to a story: - The video should replace the image in the story detail view - Company standard web video controls should be displayed - The video should be of the highest quality technically possible - Rights protected video needs to be geo-restricted to the UK and Ireland - Users that do not have the rights to watch a particular video should receive a message detailing the fact, "ERROR: You are outside of the permitted area. This video is only available within the UK and IRELAND." The CMS should enable an editor to: - set an expiry time for video - set geo-restrictions - Create a video image/headline/synopsis EDGE CASES:	
Epic	VIDEO ADVERTISING: As a business, we want pre-roll ads to display before videos begin playing, so that we can gain revenue	ACCEPTANCE CRITERIA When a user plays a video, the user should see: - an editorially controlled advertisment video - a video ad title - a "skip this ad" message to the bottom right, allowing the user to skip the add on clicking - a countdown timer for the ad on the bottom left of the video - when the video ad has completed , the user's chosen video should immediately play The CMS should enable an editor to: - set the "skip time" for the ad (the time after which a user can skip the ad) - upload the video that will be used for the ad EDGE CASES - if there is an error loading the ad, the user should proceed to their chosen video as soon as the error is detected	
Epic	NAVIGATION MENU: As a Sports Fan, I want a navigation menu, so that I can navigate to the pages that I am interested in	ACCEPTANCE CRITERIA The user should see a top level navigation menu at the top of each web page on the site: - the menu should have a menu item for each home page on the site - clicking on a menu item should take the user to the relevant home page for that item	

Notes:

- the BBC Sports Website has been used as an example here. There is a Content Management System (CMS) being used to publish content on the site.

- all the items are considered epics until they are broken down further.

- a lot of the acceptance criteria is vague and not tied down.

- a small phrase in caps such as SPORTS HOMEPAGE has been included in the epic title. This can later be used to group user stories into a 'theme'.

Summary

✓ Create the epics as non-detailed user stories incorporating user feedback.

✓ This is often done with the team in a product backlog refinement / creation meeting.

Step 2 - Prioritization

Every product backlog item (PBI) is prioritized based on its business value. The product owner makes a comparison between items on the backlog to determine their relative priority. Doing this sometimes tends to cause the PO to think about each item's size, business value, and any associated risk. However, the value of the item to the business is the key focus here.A business value figure is often used to carry this out. E.g a number from 1 to 10o can be used. Where 1 is the lowest value and 100 is the highest. The scale is not important

The backlog item's order of priority descends as we progress through the backlog (at the top are High-priority items, at or near the bottom are lower priority items). When choosing work for the impending sprint, the scrum team picks from the highest priority items, so that they complete the most important items first. Ensure the top backlog items are "ready": feasible, clear, and testable, to allow the team in turning the items into a product increment, and facilitate a realistic commitment.

Since product backlog's items are of different sizes, the small ones can be easily completed in one sprint. However, the big ones can leave the team unsure of what's involved or their implementation period. The good thing is that, as items move closer to the backlog's top, the team makes them smaller and more streamlined to make everyone understand the work to be addressed in upcoming sprints.

Similar to estimation, prioritizing the first product backlog could be quite overwhelming. One may be confused on how to ensure competing stakeholder demands are well balanced, bearing in mind the end product, costs and risks. Fortunately, there are so many techniques to make the work easier which include relative weighting and Innovation Games. No matter what prioritization technique you prefer, ensure to order the work so that the team builds features that bring the most value to the business, the customers and the stakeholders. Failure to prioritize your work increases the risk of delivering incomplete user stories as schedules and resources change and even lower priority user stories rather than higher ones.

A sample of a sports website product backlog of prioritised Epics (enlarge to

view clearly)

Epic	SPORTS HOMEPAGE: As a Sports Fan, I want to see the Top sport headlines , so that i can stay fully informed of sporting news	ACCEPTANCE CRITERIA On entry to the site, the user should be able to access a page that includes editorially controlled news headlines For each news story the user should see - an image - a title - a brief summary of the story Clicking on a story should take the user to the story detail page The CMS should enable an editor to: - Deliver images of different sizes - Order stories on the page and set the position on the page - Supply relevant meta-data (as above for the story to be represented on the homepage EDGE CASES - If there are no news stories due to an error, the user should see the appropriate error message: "ERROR: There are currently no stories available, please try again later."	
Epic	STORY DETAIL PAGE: As a Sports Fan, I want to see a story detail page, so that I can stay informed with a detailed news story	ACCEPTANCE CRITERIA The user should see: - a story title - the amount of time since the news story was published in hours - a large headline image - a brief story summary - a detailed story consisting of paragraphs of text - related stories small icon - related stories small headline All stories should update dynamically without reloading the page The CMS should enable an editor to: - Supply relevant story meta-data (as above) - Set related stories with the data above EDGE CASES - If there is no data for the story due to an error, the user should see an image placeholder and the appropriate error message: "ERROR: There is currently no story available, please try again later."	

Epic	VIDEO: As a Sports Fan, I want to see video and audio related to the current story	ACCEPTANCE CRITERIA A user should be able to click and watch a video when it is editorially assigned to a story: - The video should replace the image in the story detail view - Company standard web video controls should be displayed - The video should be of the highest quality technically possible - Rights protected video needs to be geo-restricted to the UK and Ireland - Users that do not have the rights to watch a particular video should receive a message detailing the fact: "ERROR: You are outside of the permitted area. This video is only available within the UK and IRELAND." The CMS should enable an editor to: - set an expiry time for video - set geo-restrictions - Create a video image/headline/synopsis EDGE CASES: - If there is no data for the video due to an error, the user should see an image placeholder and the appropriate error message: "ERROR: There is currently no video available, please try again later."	
Epic	CRICKET HOME PAGE: As a Cricket Fan I want to see a Cricket home page so that I can stay informed with the top cricket news stories	ACCEPTANCE CRITERIA: The user should see: - An editorially controlled news list of stories Image Links to the following: - Live Scores for all competitions including Score Cards - Schedule - Latest Results - League Tables For each news story the user should see - an image - a title - a brief summary of the story The CMS should enable an editor to: - Deliver images of different sizes - Order stories on the page and set their position on the page - Supply relevant meta-data (as above for the story to be represented on the homepage EDGE CASES - If there are no news stories due to an error, the user should see the appropriate error message should be displayed: "ERROR: There are currently no stories available, please try again later."	
Epic	FOOTBALL HOME PAGE: As a Football Fan I want to see a Football home page so that I can stay informed with the top football news stories	ACCEPTANCE CRITERIA: The user should see: - An editorially controlled news list of stories Image links to the following - Results (including team line ups, match stats, match summary info) - Live Scores (including team line ups, match stats, match summary info) - Fixtures - League Tables For each news story the user should see - an image - a title - a brief summary of the story The CMS should enable an editor to: - Deliver images of different sizes - Order stories on the page and set their position on the page - Supply relevant meta-data (as above for the story to be represented on the homepage EDGE CASES: - If there are no news stories due to an error, the user should see the appropriate error message should be displayed: "ERROR: There are currently no stories available, please try again later."	
Epic	NAVIGATION MENU: As a Sports Fan, I want a navigation menu, so that I can navigate to the pages that I am interested in	ACCEPTANCE CRITERIA The user should see a top level navigation menu at the top of each web page on the site - the menu should have a menu item for each home page on the site - clicking on a menu item should take the user to the relevant home page for that item	

Epic	TENNIS HOME PAGE: As a Tennis Fan, I want to see a Tennis home page so that I can stay informed with the top tennis news stories	ACCEPTANCE CRITERIA: The user should see: - An editorially controlled news list of stories Image Links to the following: - Results for ATP and WTA tours + majors - Season schedule for ATP and WTA - Standings ATP and WTA rankings and the WTA champions race The CMS should enable an editor to: - Deliver images of different sizes - Order stories on the page and set their position on the page - Supply relevant meta-data (as above for the story to be represented on the homepage EDGE CASES - If there are no news stories due to an error, the user should see the appropriate error message should be displayed: "ERROR: There are currently no stories available, please try again later."	
Epic	VIDEO ADVERTISING: As a business, we want pre-roll ads to display before videos begin playing, so that we can gain revenue	ACCEPTANCE CRITERIA When a user plays a video, the user should see: - an editorially controlled advertisement video - a video ad title - a "skip this ad" message to the bottom right, allowing the user to skip the add on clicking - a countdown timer for the ad on the bottom left of the video - when the video ad has completed , the user's chosen video should immediately play The CMS should enable an editor to: - set the "skip time" for the ad (the time after which a user can skip the ad) - upload the video that will be used for the ad EDGE CASES - If there is an error loading the ad, the user should proceed to their chosen video as soon as the error is detected	

Notes:

- The Product Owner (PO) has decided that getting the football (soccer) home page done early is of higher value to the business than the other sports because it is the most popular sport in the world according to research.

- Similarly the Navigation Menu is higher value to get done early, because without it, the site cannot be navigated

- The other sports home pages are similar to football and therefore should be easier to build after the football page has been completed

- The PO has decided that the sports pages, navigation, story and video pages form the core of the site and meet the business vision in a worst case scenario.

- Advertising is important, but revenue is a goal for another release, so the business would launch without it if push came to shove. This release is about brand awareness!

Here is a summary of what we've learnt in this step.

In Summary

✓ Prioritize the backlog as epics based each item's business value

and balanced against risks.

✓ Put high priority items at or near the top and lower priority items at or near the bottom of the product backlog.

✓ Keep your backlog items focused and as concise as possible (eliminate features clouding your Product)

Step 3 -Estimate The Product Backlog

Estimating generally involves the scrum team making relatively precise estimates of the product backlog items. Estimation is a little complicated but it gets better with time, practice and team participation. First and foremost, the entire team needs to collaborate in estimating the product backlog items. During estimation, your team gives each story a relative size estimate represented in an abstract unit of measurement (usuallywhat is known as story points). Estimates are important because they:

- ✓ Help in answering the question" when will we be done"
- ✓ Assist the PO in determining the priority of each item

Such estimates provide the PO and the team with an idea of each story's cost and consequently helps the product owner in prioritizing stories corresponding to each other. Items with over-stretched estimates are expensive to implement in terms of resources and time. Therefore, high-cost items on the product owner's wish list decrease in priority. The PO must group the backlog's items into long-term and near-term and the team should focus on estimating the near-term ones first. Longer term items can be rather vague, though they should be roughly estimated to help prioritize them. The word "rough" estimates will change once the team fully understands and starts working on the long-term items.

To estimate your user stories and create story points for them; two known techniques your team can use are wall estimation and planning poker, but, there're many others available. This again is often done with the team in a product backlog refinement meeting.

In short, with the planning poker method,

2. the product owner reads out the story and acceptance criteria.
3. The team discuss and ask questions.
4. each team member holds up a planning poker card with an estimate of effort. The planning poker cards have values 1,2,3,5,8,13,20,40 and 100.

The larger the number, the larger the estimate. This is a measure of effort!

5. The two team members with the largest difference in values discuss the reason for their difference for a short time.
6. the whole team re-estimate, each showing their planning poker card again
7. When a consensus is reached, the story point value agreed is written on the story card. The team then move on to the next story.

A sample of a sports website product backlog of estimated Epics (enlarge to view clearly)

Issue Type	Summary	Description	Website Design Concept	Estimates (Story Points)
Epic	SPORTS HOMEPAGE: As a Sports Fan, I want to see the 'top sport headlines', so that I can stay fully informed of sporting news	ACCEPTANCE CRITERIA On entry to the site, the user should be able to access a page that includes editorially controlled news headlines. For each news story the user should see: - an image - a title - a brief summary of the story Clicking on a story should take the user to the story detail page. The CMS should enable an editor to: - Deliver images of different sizes - Order stories on the page and set the position on the page - Satisfy relevant meta data (as above for the story to be represented on the homepage) EDGE CASES - If there are no news stories due to an error, the user should see the appropriate error message: "ERROR: There are currently no stories available, please try again later."		40
Epic	STORY DETAIL PAGE: As a Sports Fan, I want to see a 'story detail' page, so that I can stay informed with a detailed news story	ACCEPTANCE CRITERIA The user should see: - a story title - the amount of time since the news story was published in hours - a large headline image - a brief story summary - a detailed story consisting of paragraphs of text - related stories small icon - related stories small headline All stories should update dynamically without reloading the page. The CMS should enable an editor to: - Supply relevant story meta data (as above) - Set related stories with the data above EDGE CASES - If there is no data for the story due to an error, the user should see an image placeholder and the appropriate error message: "ERROR: There is currently no story available, please try again later."		40

Epic	VIDEO: As a Sports Fan, I want to see video and audio related to the current story	ACCEPTANCE CRITERIA A user should be able to click and watch a video when it is editorially assigned to a story. - This video should replace the image in the story detail view - Company standard web video controls should be displayed - The video should be of the highest quality technically possible - Rights protected video needs to be geo-restricted to the UK and Holland - Users that do not have the rights to watch a particular video should receive a message detailing the fact: "ERROR: You are outside of the permitted area. This video is only available within the UK and IRELAND." The CMS should enable an editor to: - set an expiry time for video - set geo-restrictions - Create a video image/headline/synopsis EDGE CASES - If there is no data for this video due to an error, the user should see an image placeholder and the appropriate error message: "ERROR: There is currently no video available, please try again later."		100
Epic	CRICKET HOME PAGE: As a Cricket Fan I want to see a Cricket home page so that I can stay informed with the top cricket news stories	ACCEPTANCE CRITERIA: The user should see: - An editorially controlled news list of stories Image Links to the following: - Live Scores for all competitions including Score Cards - Schedule - Latest Results - League Tables For each news story the user should see: - an image - a title - a brief summary of the story The CMS should enable an editor to: - Deliver images of different sizes - Order stories on the page and set their position on the page - Supply relevant meta data (as above) for the story to be represented on the homepage EDGE CASES - If there are no news stories due to an error, the user should see the appropriate error message should be displayed: "ERROR: There are currently no stories available, please try again later."		40
Epic	FOOTBALL HOME PAGE: As a Football Fan I want to see a Football home page so that I can stay informed with the top football news stories	ACCEPTANCE CRITERIA: The user should see: - An editorially controlled news list of stories Image links to the following: - Results (including team line ups, match stats, match summary info) - Live Scores (including team line ups, match stats, match summary info) - Fixtures - League Tables For each news story the user should see: - an image - a title - a brief summary of the story The CMS should enable an editor to: - Deliver images of different sizes - Order stories on the page and set their position on the page - Supply relevant meta data (as above) for the story to be represented on the homepage EDGE CASES - If there are no news stories due to an error, the user should see the appropriate error message should be displayed: "ERROR: There are currently no stories available, please try again later."		40
Epic	NAVIGATION MENU: As a Sports Fan, I want a navigation menu, so that I can navigate to the pages that I am interested in	ACCEPTANCE CRITERIA: The user should see a top level navigation menu at the top of each web page on the site - the menu should have a menu item for each home page on the site - clicking on a menu item should take the user to the relevant home page for that item		8

Epic	TENNIS HOME PAGE: As a Tennis Fan, I want to see a Tennis home page so that I can stay informed with the top tennis news stories	ACCEPTANCE CRITERIA: The user should see - An editorially controlled news list of stories Image Links to the following: - Results for ATP and WTA tours + majors - Season schedule for ATP and WTA - Standings ATP and WTA rankings and the WTA champions race The CMS should enable an editor to: - Deliver images of different sizes - Order stories on the page and set their position on the page - Supply relevant meta data (as above for the story to be represented on the homepage) EDGE CASES - If there are no news stories due to an error, the user should see the appropriate error message should be displayed: "ERROR: There are currently no stories available, please try again later."		40
Epic	VIDEO ADVERTISING: As a business, we want pre-roll ads to display before videos begin playing, so that we can gain revenue	ACCEPTANCE CRITERIA When a user plays a video, the user should see: - an editorially controlled advertisment video - a video ad title - a "skip this ad" message to the bottom right, allowing the user to skip the add on clicking - a countdown timer for the ad on the bottom left of the video - when the video ad has completed , the user's chosen video should immediately play The CMS should enable an editor to: - set the "skip time" for the ad (the time after which a user can skip the ad) - upload the video that will be used for the ad EDGE CASES - If there is an error loading the ad, the user should proceed to their chosen video as soon as the error is detected		40

Notes:

- Most of the epics have high estimates of 40 or 100. This can be because we are in the early stages of the product and criteria is vague. Also because there is alot to do due to the effort required to build a CMS and a website.

- The Navigation story is much smaller than the others meaning it is less effort. It has been called an epic for now because the team do not know if they can accomplish it in a sprint. We will think about that in another product backlog refinement meeting (coming up later in this class).

Let's now take a quick look at a summary of what we've learnt in this chapter.

Summary

✓ The team need to estimate the backlog, preferably in priority order.

✓ Consider using the estimation techniques suggested but feel free to use others.

✓ Remember that estimates will likely be large until the epics have been refined to stories that can fit in a sprint.

Step 4 - Refining and Grooming Your Product Backlog

When a story is added, the PO makes preliminary assessment of the story's priority based on how he understands it. To better understand, he talks to the story's creator so that he's able to answer any question that may arise. The PO himself is often also the creator. At a set time during each sprint, the PO meets the team to discuss new stories with the intention that he'll be able to prioritize them accurately relative to the other backlog items. This process is known as refining or grooming the backlog.

Product backlog grooming or refinement is an activity that keeps your backlog up to date and therefore must happen frequently. If grooming is done correctly, you'll develop a successful product that benefits the user and the customer plus the organization. As the image below shows, your backlog could change based on knowledge obtained from developing product and exposure to users, customers, and other stakeholders.

Grooming your backlog enables you to integrate your newest insights into the backlog and this guarantees that you'll develop the right product properly. And that the new product backlog has plenty of ready items for the next sprint.

Steps in backlog grooming:

- ✓ Analyze the user , customer and team feedback
- ✓ Integrate new knowledge
- ✓ Decide your next course of action
- ✓ Prepare the backlog; create ready, more detailed user stories.

Taking the steps above provides a deep backlog. A concise backlog lets you integrate gained insights and a visible one promotes creative conversations among persons involved in the development effort. For easy access, put your backlog up on the wall or on a wiki server.

Each sprint, the PO and development team ought to spend 5%-15% of their time on grooming activities. The team should understand what's upcoming and

changing to be able to break down any epics that are increasing priority, estimate any new stories, and do some planning and emergent design for upcoming stories. For this to happen, during each sprint planning meeting, the PO and the team should set aside some time for grooming their backlog. Grooming can happen before starting new development work or during the next spring. If you need customer and user feedback, ensure to get the relevant data, analyze it then integrate these latest insights into your backlog prior to coding.

For a two-week sprint, I prefer to hold this meeting during its second week. This way, the product owner gets ample time to have meaningful conversations with the stakeholders and customers, understand business changes, and to clarify user stories and prioritize new or rising stories. It's also a good time in a sprint, to start anticipating coming sprints and even choose what time to hold the meeting. Allowing adequate time to complete grooming activities is essential.

In a normal grooming meeting, the PO shows what's new, what's changed, and the next few sprints plan. Besides, estimating new stories and chunking big ones that need to be completed soon, the team sets some time in this meeting to review the system's current architecture and to design upcoming features. In the course of this, story estimates often change, and new stories pop up while epics are chucked into minor stories.

Themes

One technique often used is to group related stories into a "theme". A theme is a collection of related stories, grouped together similar to placing a rubber band around them. I usually achieve this with a phrase in caps such as "VIDEO STORY" or "FOOTBALL HOME PAGE". All stories with this prefix are related to this theme.

It takes work

It may seem like a piece of cake but it can be quite tasking. The PO must be ready to answer any questions arising. Conflicts can arise if, for instance, the PO starts planning the next sprint stories without making it clear to the team so

that they can make good estimates. If you experience such tension during sprint planning meetings, the Scrum Master must coach the PO on the information he needs to gather from the customers and stakeholders and bring to the team.

At the end of each refinement meeting, the PO must communicate the changes to the stakeholders and customers so that everybody can see what's new, what's on the cards, and the updated release plan.

A sample of a sports website product backlog of refined user stories for sprint (enlarge to view clearly)

User Story	STORY DETAIL PAGE: As a Sports Fan, I want to see a 'story detail' page, so that I can stay informed with a detailed news story	ACCEPTANCE CRITERIA: The user should see: - a story title - the amount of time since the news story was published in hours - a large headline image - a brief story summary - a detailed story consisting of paragraphs of text - related stories small icon - related stories small headline All stories should update dynamically without reloading the page The CMS should enable an editor to: - Supply relevant story meta-data (as above) - Set related stories with the data above EDGE CASES: - If there is no date for the story due to an error, the user should see an image placeholder and the appropriate error message: 'ERROR: There is currently no story available, please try again later.'		20
User Story	STORY DETAIL PAGE: As a Content Editor, I want a CMS screen so that I can publish content to a 'story detail' page	ACCEPTANCE CRITERIA: The content editor should be able to enter: - a story title - the amount of time since the news story was published in hours - a large headline image - a brief story summary - a detailed story consisting of paragraphs of text - related stories small icon - related stories small headline The CMS should enable an editor to: - Supply relevant story meta-data (as above) - Link previously added stories to the story		20

User Story	SPORTS HOMEPAGE: As a Sports Fan, I want to see the 'top sport headlines', so that I can stay fully informed of sporting news	ACCEPTANCE CRITERIA: On entry to the site, the user should be able to access a page that includes editorially controlled news headlines For each news story the user should see - an image - a title - a brief summary of the story Clicking on a story should take the user to the story detail page EDGE CASES: - If there are no news stories due to an error, the user should see the appropriate error message: 'ERROR: There are currently no stories available, please try again later.'		20
User Story	SPORTS HOMEPAGE: As a Content Editor, I want a CMS screen so that I can publish 'top sport headlines' to the sports homepage	ACCEPTANCE CRITERIA: The editor should be able to enter (for up to 20 stories - an image - a title - a brief summary of the story The CMS should enable an editor to: - Deliver images of any size in 16:9 ratio - Order stories on the page and set the position on the page - Supply relevant meta-data (as above for the story to be represented on the homepage		20

User Story	FOOTBALL HOME PAGE: As a Football Fan I want to see a Football home page so that I can stay informed with the top football news stories.	ACCEPTANCE CRITERIA: The user should see: - An editorially controlled news list of stories Image links to the following: - Results (including team line ups, match stats, match summary info) - Live Scores (including team line ups, match stats, match summary info) - Fixtures - League Tables For each news story the user should see: - an image - a title - a brief summary of the story EDGE CASES: - If there are no news stories due to an error, the user should see the appropriate error message should be displayed. "ERROR: There are currently no stories available, please try again later."		20
User Story	FOOTBALL HOME PAGE: As a Content Editor I want a CMS page so that I can publish football stories to the football news page	ACCEPTANCE CRITERIA: The Content Editor should be able to add for each editorially controlled story: Image links to the following: - Results (including team line ups, match stats, match summary info) - Live Scores (including team line ups, match stats, match summary info) - Fixtures - League Tables For each news story the content editor should be able to add: - an image - a title - a brief summary of the story The CMS should enable an editor to: - Deliver images of different sizes - Order stories on the page or it will their position on the page - Supply relevant meta-data (as above) for the story to be represented on the homepage		20
User Story	NAVIGATION MENU: As a Sports Fan, I want a navigation menu, so that I can navigate to the pages that I am interested in	ACCEPTANCE CRITERIA: The user should see a top level navigation menu at the top of each web page on the site: - the menu should have a menu item for each home page on the site - clicking on a menu item should take the user to the relevant home page for that item		8
User Story	VIDEO: As a Sports Fan, I want to see video and audio related to the current story	ACCEPTANCE CRITERIA: A user should be able to click and watch a video when it is editorially assigned to a story: - The video should replace the image in the story detail view - Company standard web video controls should be displayed - The video should be of the highest quality technically possible - Rights protected video needs to be geo-restricted to the UK and Ireland - Users that do not have the rights to watch a particular video should receive a message detailing the fact. "ERROR: You are outside of the permitted area. This video is only available within the UK and IRELAND." EDGE CASES: - If there is no data for the video due to an error, the user should see an image placeholder and the appropriate error message. "ERROR: There is currently no video available, please try again later."		20

Epic	CRICKET HOME PAGE: As a Cricket Fan i want to see a Cricket home page so that I can stay informed with the top cricket news stories	ACCEPTANCE CRITERIA The user should see: - An editorially controlled news list of stories Image Links to the following: - Live Scores for all competitions including Score Cards - Schedule - Latest Results - League Tables For each news story the user should see - an image - a title - a brief summary of the story The CMS should enable an editor to: - Deliver images of different sizes - Order stories on the page and set their position on the page - Supply relevant meta-data (as above) for the story to be represented on the homepage EDGE CASES - If there are no news stories due to an error, the user should see the appropriate error message should be displayed: "ERROR: There are currently no stories available, please try again later."		40
Epic	TENNIS HOME PAGE: As a Tennis Fan, i want to see a Tennis home page so that I can stay informed with the top tennis news stories	ACCEPTANCE CRITERIA The user should see: - An editorially controlled news list of stories Image Links to the following: - Results for ATP and WTA tours + majors - Season schedule for ATP and WTA - Standings ATP and WTA rankings and the WTA champions race The CMS should enable an editor to: - Deliver images of different sizes - Order stories on the page and set their position on the page - Supply relevant meta-data (as above) for the story to be represented on the homepage EDGE CASES - If there are no news stories due to an error, the user should see the appropriate error message should be displayed: "ERROR: There are currently no stories available, please try again later."		40
Epic	VIDEO ADVERTISING: As a business, we want pre-roll ads to display before videos begin playing, so that we can gain revenue	ACCEPTANCE CRITERIA When a user plays a video, the user should see: - an editorially controlled advertisment video - a video ad title - a "skip this ad" message to the bottom right, allowing the user to skip the add on clicking - a countdown timer for the ad on the bottom left of the video - when the video ad has completed , the user's chosen video should immediately play The CMS should enable an editor to: -set the "skip time" for the ad (the time after which a user can skip the ad) -upload the video that will be used for the ad EDGE CASES - If there is an error loading the ad, the user should proceed to their chosen video as soon as the error is detected		40
User Story	VIDEO: As a Sports Fan, I want to see video and audio related to the current story with GEO RESTRICTION	ACCEPTANCE CRITERIA A user should be able to click and watch a Geo Restricted video when it is editorially assigned to a story: - Rights protected video needs to be geo-restricted to the UK and Ireland - Users that do not have the rights to watch a particular video should receive a message detailing the fact : "ERROR: You are outside of the permitted area . This video is only available within the UK and IRELAND."		20
Epic	VIDEO: As a Video Editor, I want to see a CMS page so I can publish video and audio related to a particular story	ACCEPTANCE CRITERIA A Video Editor should be able to: - Upload a video for a particular story The CMS should enable an editor to: -set an expiry time for video -set geo-restrictions - Create a video image/headline/synopsis		40

Notes:

- The stories highlighted in yellow are no longer epics. They have been split if necessary and refined to a size that can be completed by the team in one sprint

- In this case, the CMS aspect of each story has been separated out into it's own story. This was a team decision. However usually I like to keep back-end and front-end tasks in one story where possible for end-to-end testing reasons. This is not always easy.

- Related stories have been grouped using the phrase in caps. This is known as a "theme".

- The 20 point stories still may be too large for the team to finish easily in a sprint. They may want to break them down further in another refinement meeting.

- The team now have enough to take at least one or more stories into a sprint planning session.

- Further backlog refinement meetings can be used to break down the remaining epics or even further break down the stories.

To wrap this up, let's take a quick look at a summary of what we've learnt so far.

Summary

Update your product backlog to incorporate insights gained from releasing or demoing product to the users and customers.

✓ Do your grooming (or refinement) before or during development depending on whether you need user or customer feedback to continue developing your product.

✓ Ensure the PO gets the development team on board during

grooming to tap into their creativity and knowledge. Include stakeholders as appropriate.

✓ The top stories in the product backlog (needed for a sprint) should be ready for a team to take into sprint planning and small enough to be completed in a sprint.

We are now going to discuss some tips for grooming your product backlog (refinement) to make it manageable.

Tips & Strategies

The following are some fine-tuning tips for managing your product backlog. In my tips, I'll try to articulate very effective ways to execute the Scrum framework. My tips are consistent with scrum principles and are based on my abundant Scrum coaching experience and research:

Tip #1: PO Should Know That He/She Will Be Expected To Present Enough Work That Can Last About Two Sprints Beyond The Current Sprint During All The Backlog Refinement Sessions For The Current Sprint

The rationale behind bringing this much work to the meeting is:
1. Frequently user stories are de-prioritized based on feedback in the meeting, so you need some leftover work set to fill a sprint.

2. Having some finely refined user stories away from what you're working on in a Sprint is good practice anyway- someone might free up and need more work.

To know the work involved in 2 Sprints, the PO needs to collaborate with the team, but this doesn't need to be a lengthy discussion -- merely a rough guess. With a bit of luck, your team could've seen user stories at a Release Planning meeting or in any other backlog refinement conversations.

Tip #2: Avoid Scheduling Backlog Refinement During The First Or Last 20% Of The Sprint

During the Sprint's first 20%, the team is just getting started working on stuff, and needs room to start smoothly. And for the last 20%, they're working hard to complete their current sprint items, so it's also not the best time. Do backlog refinement during the middle 60% of the sprint.

Tip #3: The Backlog Items Must Be Fine Grained And Properly Understood By The PO For This Meeting To Flow Well. Define Initial Set Of Acceptance Tests Before The Meeting Occurs

When the PO brings a backlog item to the meeting, all you want to hear is “I've worked on this item in collaboration with the customers and and I’ve a really good idea of thebacklog items plus their acceptance tests (acceptance tests are the criteria for the customer’s acceptance of the story).Such solid information is enough to start working almost immediately!

Although the initial acceptance tests could be high level, it’s better than zero acceptance tests. The team and PO should focus on getting good detailed acceptance tests.

Tip #4: Treat Every Backlog Refinement Meeting Just Like The Beginning Of Sprint Planning Meeting

Talk about what will end up being developed in the forthcoming sprint. The team should ask for clarifications from the PO and estimate the items. The PO might find the need to change priorities. Collaboration is important.

Tip#5: Everyone Should Understand That Estimates Are Provisional Until A User Story Is Accepted Into A Sprint

Given that the team are previewing the user stories, let them know that they don't need to stress out a lot on the estimates. Inform them that a user story's estimate is not final and that the backlog item is accepted into the sprint. If someone gets new info amid the backlog refinement and the Sprint Planning Meeting, they're free to discuss and re-estimate the user stories.

Tip #6: Strive To Optimize Your Time During The Meeting

Discuss and promptly estimate the well defined PBI's. Only revisit the previously discussed items if there is new information about them. For user stories with new information, just discuss the new information enough to estimate them afresh.

Tip #7: Remember That Backlog Items Are Actually A Collaboration Between The PO And The Team

The Product Owner holds responsibility to ensure that user stories are sufficiently detailed for the development team's use. However, that doesn't mean that the PO does all that work; he/she should collaborate with the dev team and also probably with customers or end users. It's perfectly okay, and actually encouraged, that there is a meeting between PO and some of the team members to help detail a user story, before the grooming session ("informal backlog refinement"). Hold the necessary discussions to ensure you present a well thought out, well detailed user story at the refinement session. Presenting a user story that is new to all developers at a backlog refinement session, could be a lot slower. Just let the team join in earlier.

Tip #8: Keep Your Eye On The Goals Of The Meeting

One of the goals is that when the meeting ends, you'll have a handful of user stories lined up and ready for the Sprint, and even enough for two sprints to come. Every major question answered (or assigned to someone to answer), plus, the team feels knowledgeable and confident about the upcoming work. You may not know all the details but you've got something. At the end of the meeting, the team should be asked if there're any identified risks- someone should be assigned to mitigate it before the upcoming sprints.

Tip #9: Get Volunteers For Action Items For Any Unknowns Or Big Risks

The PO should take the big requirements issues and the development team assigned the big technology risks or unknowns to research on. The assigned persons needs to deliver the report on the action item at or before the next refinement meeting or else before (or at) the Sprint Planning Meeting. Normally, the item will be re-estimated based on the new information. To maintain self organization, a person in the team should volunteer to take an action item. To help, the Scrum Master can ask questions like "Who can take this item and give us feedback?" Nobody should dominate in assigning the action items, they should be voluntary.

Tip #10: You're Definitely Free To Split User Stories During This Meeting

Whoever feels the need to split items, can do so during a refinement session. It's fine to re-estimate them too. Re-estimate the split items as if they were new, independent, user stories.

Tip #11: Everyone Should Understand That Product Backlog Order Is Provisional Until A User Story Has Been Accepted Into A Sprint

If need be, the product owner is free to shuffle up the order of things between sprint planning meeting and the backlog refinement process even if it is not common to make any changes at the time.

Tip #12: Be Brave To Discuss A Couple Of Items Farther Down The Backlog

The need to discuss items further down the backlog could be because:

- ✓ The PO needs estimates to understand the user stories' approximate size.
- ✓ The dev team wants to see external dependencies that must be started on ASAP as they wait for the other item's details to be flushed out.

Of course, you shouldn't form a habit of doing Big Requirements Up Front (BRUF) or Big Design UP Front (BDUF), but looking to items at the bottom can be advantageous to the organization in many circumstances. Don't be afraid to do it, but take care not to slip back into BDUF and BRUF.

Tip #13: LetThe Dev Team Review The User Stories To Be Discussed About 24 Hours Before The Refinement Session

The dev team, should be reminded to review any documented user stories before any backlog refinement session (such as on index cards or a wiki), so that they prepare to speak intelligently about them.

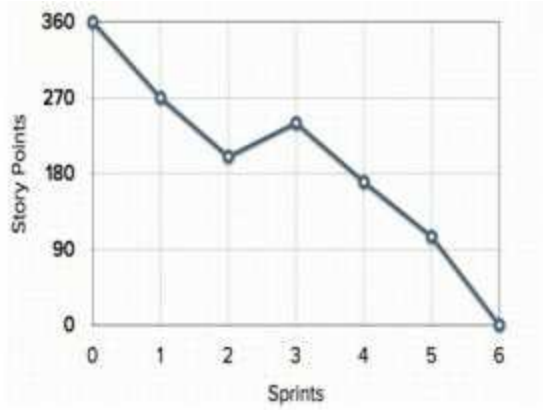
If the PO makes tons of last minute changes right up until the refinement session, a great deal of the information is seen for the first time in the meeting, possibly making the meeting take longer and be more confusing. Ideally, it helps if the PO presents the edits for review 24 hours prior to the backlog refinement session.

Tip #14: Have A Mini Backlog Refinement Prior To Holding The Backlog Refinement For The Full Team

In an Informal Product Backlog Refinement, the PO usually works with some stakeholders (if appropriate) and development team members in refining stories and their place within the backlog. It can be a daily, if not hourly, activity for the Product Owner. However the early collaborators should seek to avoid anchoring or skewing the whole Dev Team's estimates. It is important to remember that this is just a clue to the product owner to help get the stories ready for the refinement meeting with the whole dev team. Estimates should only be captured from the whole dev team.

Tip #15: Track Progress Using Release Burndown Charts

You can track your progress in a scrum project using the release burndown chart, which is updated by the scrum master when each sprint ends. The X (horizontal) axis of the chart shows the sprints while the vertical axis shows the amount of work that's left over at the time of starting each sprint. Remaining work can be shown in whatever unit preferred by the team -- story points, team days, ideal days, and so on.



The burndown chart shown above indicates that the team started a project that had six sprints. The team started with 360 story points of work. They planned to do 60 points per sprint for them to complete the work within six sprints. The first sprint took off fine and they completed 90, and were left with 270.

Things also progressed well during the second sprint. However, during the third sprint, the remaining estimated work really burned up. The reason for this could be that there was work added or because some estimates of the remaining work were changed. Afterwards, things again went well.

This kind of release burndown chart is applicable for many teams and in many situations. However, for projects that have lots of changing requirements, there are other alternative release burndown charts that you can use to keep your agile scrum project on track. The burndown chart is important for any agile project and helps the team clearly see what's going on and the progress made each sprint.

Tip #16: Feel Free To Introduce Late Breaking User Stories. As You Do That, Strive To Minimize Them But Also Embrace Them As And When They Happen

The idea here is to remain agile. Your major goals that revolve around doing backlog refinement include reducing risks and unknowns (although you should keep in mind that you cannot identify all risks beforehand). Keep in mind that backlog refinement doesn't necessarily eliminate risks but only minimizes them. And while late breaking user stories may still happen (although fairly rarely), you should try to welcome them in the team. Keep in mind that if many of your backlog items are late breaking, this is a bad sign.

Tip #17: Use One Issue Tracker and One Backlog!

You should strive not to use multiple systems for tracking bugs, requirements and different work items. On the contrary, you should aim to keep a single backlog for work for the development team. Multiple trackers and backlogs can lead to a situation where you encounter duplicates and inaccuracies. All in all there should be one version of the truth to avoid problems down the line.

Tip #18: Watch Out For Too Many Changes And Limit Your Scope

When work is in progress, ensure to keep any changes to a minimum since regular changes may actually end up disrupting the development team and affect flow, focus and morale. As such, once the backlog has grown beyond your team's capacity in the long term, you should consider closing the issues that the team will probably never get to.

While it may be tempting to continuously add new stories to the backlog just to ensure you don't say no to your stakeholders, this can take up too much time if you want to keep your backlog up-to-date. Otherwise, it won't be fair to simply say 'sure, I will add that to the backlog' while you are thinking 'I know we will definitely not build this any time soon'.

In addition, you should try to flag the issues that have a specific resolution such as 'out of scope' in your team's issue tracker to help you with research later.

It is important to watch out for some of the following anti-patterns:

- ✓ Where the product owner prioritizes backlog at the beginning of a project but fails to adjust it as new feedback rolls in from the different stakeholders and developers
- ✓ The backlog is usually stored as a document that's stored locally but shared very infrequently thus preventing any interested parties from getting the latest updates.

Tip #19: Ensure To Experiment With The Amount Of Refinement, Which Your Team Does

From the onset, until you get to the point of 'two sprints ahead' (i.e. beyond your current sprint), you will definitely require more refinement than what's normal. As such, you should try to experiment with how much refinement that you actually need to end up staying "2 sprints ahead" after you get caught up. For a start, you may want to aim at typically 2 to 4 hours a week until you are '2 sprints ahead'. However, don't let these numbers depress you. Instead, simply use them as a starting point so that you can adjust your time up or down as necessary. Nonetheless, keep in mind that scrum guidance for refinement is actually 10% of the development teams time per sprint. This means that if you have a 2 week sprint, you need 1 full day (about 8 hours) per sprint. After that, your team should be able to stay under that time box after you are '2 sprints ahead'. This is a guideline that should help you greatly!

Tip #20: Ensure To Retrospect, Inspect And Then Adapt

Like any other practice, you should ensure that the team inspects and then adapts its practice. For instance, if your team feels that backlog refinement ends up wasting too much time unnecessarily, then you need to work towards removing the impediments. You may want to come up with a list of five reasons why the problem exists then read some bit about good backlog refinement techniques to help you get started.

Tip #21: Transparentize, Prioritize And Stick To The Product Vision

You should strive to ensure that you do not use a complicated tool to hide the backlog. Instead, strive to make it as transparent to stakeholders as is practically possible just to make sure that everyone can check the last status and give feedback. In so doing, this will offer the much needed support in building a strong foundation of trust, which is necessary when for instance you are making difficult decisions.

As you do that, you should strive to order the product backlog depending on priority, dependencies, risk, most of all business value. You can also bear in mind how much you stand to learn as a factor for prioritization. While value and priority are often considered to be the most important (e.g. you are likely to want a house to provide shelter but that doesn't mean that you shouldn't build the foundation and the walls). As such, whenever you are ordering a backlog, ensure to take into account the risk, the dependencies and the amount of learning needed.

And finally, when all is said and done, you should stick to the vision of the product when implementing anything. Don't make changes without considering the product vision.

Conclusion

When starting any scrum project, the vision must be well understood by the entire scrum team and relevant stakeholders. A great overarching goal will help you come up with a concise clear backlog that will help ensure that your end product has the features identified in your conversations with users, customer and other stakeholders as defined in your user stories. Creating and maintaining a good product backlog requires the PO to be actively engaged with both the customer group/stakeholders and the team regularly-each sprint. Obviously, without an updated product backlog, your project will fail. As a PO, take your role seriously and keep the backlog in good condition-your customers will thank you. Once you master the concepts and practices in agile scrum, you apply them in any project that has rapid changing requirements.

Thank you again for taking this class!

I hope this class was able to help you to improve and master agile scrum retrospectives.

The next step is to start putting into practice what you have learned. All the greatest practitioners achieved their goals by putting tips such as these into practice and learning from their experiences. You can do it too!

Finally, if you enjoyed this class, would you be kind enough to leave a review?

Click here to leave a review for this!

Thank you and good luck!

Preview of 'The Scrum Master Mega Pack'

This book is an amalgamation of the following books sold as a cost effective box set (so be sure not to buy them again):

1. Complete Overview of Scrum - The Power of Scrum, In the Real World, For the Agile Scrum Master, Product Owner, Stakeholder and Development Team

2. 72 Reasons Why Scrum Works, For the Agile Scrum Master, Product Owner, Stakeholder and Development Team

3. The Scrum Checklist, For the Agile Scrum Master, Product Owner, Stakeholder and Development Team

4. Scrum of Scrums, Agile Programme Management, For the Agile Scrum Master, Product Owner, Stakeholder and Development Team

5. Scrum Top Tips, For the Agile Scrum Master, Product Owner, Stakeholder and Development Team

6. How to Become a Scrum Master, In 7 Simple Steps, For the Agile Scrum Master, Product Owner, Stakeholder and Development Team

7. How to Meet a Project Deadline with Scrum, In 7 simple steps For the Business, Agile Project Manager, Scrum Master, Product Owner, and Development Team

BONUS

8. Kanban, The Kanban guide, For the Business, Agile Project Manager, Scrum Master, Product Owner and Development Support Team

Here is a piece from the book:

6 reasons why SPRINT PLANNING works

Background

The sprint planning session is a meeting in which the team plan and commit to the stories that they will work on in a sprint. The meeting lasts no more than four hours for a two-week sprint. There are two halves to the meeting, the "what" and the "how". In the first half of the meeting, the product owner presents the list of features that he would like the team to deliver from the product backlog. He explains them and the team ask questions. In the second half of the meeting, the team break the stories into technical tasks and estimate them. The meeting ends with a commitment from the team to complete the sprint backlog within the sprint.

Reasons

1. The planning process gives solid understanding: As opposed to other methods, it is the act of discussing and understanding the proposed sprint backlog that gives the team a solid understanding of what they are building. There is no reliance on a document to do this.

2. Expert estimates: The estimates given at planning are from the most reliable source possible – those doing the work, as opposed to a single team lead.

3. Reliable estimates: Rather than individual estimates, the shared view of the team is reflected in the estimates. This is a more reliable view since it takes into account all perspectives.

4. Motivating, enjoyable experience: Planning gives a chance for people to interact and to some extent socialize. This is usually seen as an enjoyable experience as long as it is strictly timed and especially if planning poker is used as a technique (doughnuts also help greatly).

5. Product closer to customer desires: The product owner brings the business view to the session and the team bring the "builder's" view. This results in a

product much closer to what the customer desires.

6. Strict time-box keeps morale high: The time-boxed session must run to a strict time scale. A further session can be planned if absolutely necessary, but this strictness prevents fatigue.

Click here to check out the rest of (Scrum Master Mega Pack) on Amazon.

Or go to:http://www.amazon.com/dp/B00988FMU8

Check Out My Other Books

Below you'll find some of my other popular books that are popular on Amazon and Kindle as well. Simply click on the links below to check them out. Alternatively, you can visit my author page on Amazon to see other work done by me.

The Power of Scrum, In the Real World, For the Agile Scrum Master, Product Owner

Confessions of a Scrum Master, for the Agile Scrum Master, Product Owner, Stakeholder and Development Team

Kanban, The Kanban guide, For the Business, Agile Project Manager, Scrum Master, Product Owner and Development

How to Become a Scrum Master in 7 Simple Steps (Agile Project Management)

Scrum of Scrums, Agile Program Management, For the Agile Scrum Master, Product Owner, Stakeholder and Development

Scrum, The Complete Overview and Guide (Boxset), For the Agile Scrum Master, Product Owner, Stakeholder and Development

The Scrum Checklist, For the Agile Scrum Master, Product Owner, Stakeholder and Development Team

How to Meet a Project Deadline with Scrum In 7 simple steps For the Business, Agile Project Manager, Scrum Master

Scrum Top Tips, For the Agile Scrum Master, Product Owner, Stakeholder and Development Team

Selling Scrum to the Business: 72 Reasons Why Scrum Works, For the Agile Scrum Master, Product Owner, Stakeholder

Scrum, (Mega Pack), For the Agile Scrum Master, Product Owner, Stakeholder and Development Team

If the links do not work, for whatever reason, you can simply search for these titles on the Amazon website to find them.

Bonus: Subscribe to Download the FreeScrumEbook and Bonuses

When you downloadthe FreeScrumEbookand subscribe to get the free bonuses via email, you will get free access to a some of exclusive subscriber-only resources. All you have to do is enter your email address to the right to get instant access.

These exclusive bonuses and resources will help you get more out of agile scrum – to be able to reach your agile scrum goals quicker and faster than ever before like the pros. I'm always adding new bonuses and discounts as well, which you will be notified of as a subscriber. **These will help you to achieve your goals as quickly as possible!**

Here are the details of what you'll learn:

The Number one Reason why projects succeed or failwhich is key to your success going forward.

How Agile relates to Scrumone of the most confused topics in industry

A complete introduction to Scrum theory, the foundation of it's successas the theory will give you a strong foundation going forward

An overview of all the Scrum rulesso that you can set the correct framework for your projects

A complete understanding of all the Scrum rolesas these make scrum so powerful

An overview of all the Scrum eventsso that you have a summary of all the

practices the pros use

How you can take steps to get your project to the next levelso you know what to do next

Access to my blog including:

Regular lessons on how to improve scrum events

Discounts of up to 99% on all my scrum courses

To get instant access to this free scrum ebook and more incredible bonuses and resources, click the link below:

Click here for the FreeScrumEbook and Bonuses.

Or you can access it here: http://www.freescrumebook.com